Accepted Manuscript

A novel image encryption algorithm based on polynomial combination of chaotic maps and dynamic function generation

Meysam Asgari-Chenaghlu, Mohammad-Ali Balafar, Mohammad-Reza Feizi-Derakhshi

PII: S0165-1684(18)30370-0
DOI: https://doi.org/10.1016/j.sigpro.2018.11.010
Reference: SIGPRO 6984

To appear in: *Signal Processing*

Received date: 13 May 2018
Revised date: 20 October 2018
Accepted date: 14 November 2018

Please cite this article as: Meysam Asgari-Chenaghlu, Mohammad-Ali Balafar, Mohammad-Reza Feizi-Derakhshi, A novel image encryption algorithm based on polynomial combination of chaotic maps and dynamic function generation, *Signal Processing* (2018), doi: https://doi.org/10.1016/j.sigpro.2018.11.010

A novel image encryption algorithm based on polynomial combination of chaotic maps and dynamic function generation

Meysam Asgari-Chenaghlu[a], Mohammad-Ali Balafar[a], Mohammad-Reza Feizi-Derakhshi[a]

[a] *Department of Computer Engineering, Faculty of Electrical and Computer Engineering, University of Tabriz, Iran*

Abstract

Present paper introduces a polynomial combination of one dimensional chaotic maps that is blended in a dynamic image encryption algorithm. It is special because not only this combination has butterfly folding effect but also it shows generalization property over any polynomial combination. Hence, the butterfly folding effect is caused by governed parameters of polynomial combination. Moreover, multiple simulations and evaluations show the superiority of the proposed chaotic system. An application of this system, which we propose in cryptography, is a novel image encryption algorithm based on dynamic function generation. Compared to the state of the art algorithms, our image encryption algorithm has higher statistical and cryptanalytic properties. Even though this algorithm is not suitable for real-time applications such as streaming video encryption, it makes a good use of the proposed chaotic system. Uppermost cryptanalytic properties that are proven by statistical/numeric tests show good performance and reliability of proposed algorithm for image encryption tasks while unlike any other chaotic image encryption system, our algorithm uses a string input for secret key.

Keywords: Image Encryption, 1D chaotic maps, Polynomial coupling, Random number generator, Butterfly effect, Dynamic function generation

1. Introduction

This is the rapid growth of network users and data transmits that has led the security researchers to devise much secure infrastructure and cryptographic algorithms. On the other hand, the cryptanalysts have also gained more computing power to put these new inventions to test [1]. Although there are many algorithms that can be used for making a secure data transmit such as *AES* (Advanced Encryption Algorithm) [2], some successful attacks have been reported to these algorithms; *Biclique Cryptanalysis* [3] is an example. In cryptography culture, TuxECB effect is a major drawback of these conventional algorithms. Figure 1 shows an inevitable effect happening due to block cipher nature of ECB algorithms like AES. This semantically incorrect cryptographic scheme (along with other problems mentioned earlier) fuels researchers of criteria (e. g. cryptographers) towards more reliable encryption systems and algorithms.

Due to their random-like behavior and high statistical scores on various tests such as NIST random number test, correlation analysis and information entropy, chaotic image and media encryption methods have earned so much attention. However, in many aspects, successful attacks on some of these algorithms has been applied and security flaws is reported by various researchers [4–11].

In near future, with increasing higher user interests in social networking (e.g. on websites like *Facebook*, *Instagram* which mostly rely on media as content) mentioned problems can be much more critical. This happens because increase in user generated content needs much secure data transmit that can be decrypted on both sides of transmission line. However, the problem on user side rises as the decrypting device (e. g. mobile phone or a VR glass) lacks computational and storage resources. This problem can be solved by a fast, secure and easily implementable encryption algorithm that needs no more information rather than the encrypted message and a securely shared secret key. On some study cases, researchers used another image called as *key-image* for secret key to ensure this security but the employed methodology increases data transition by a factor of $\times 2$.

Considering the described demands, many chaotic and non-chaotic approaches have been proposed for image and data encryption. In the case of image and video encryption, non-chaotic and iterative cryptographic functions, such as

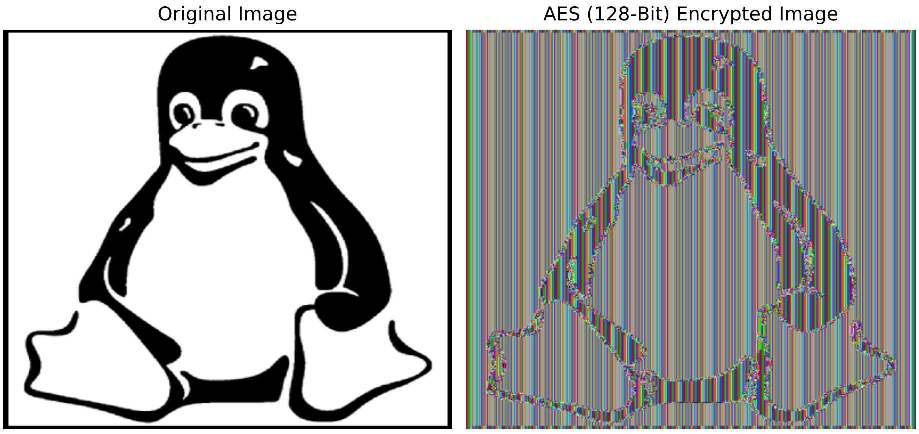

Figure 1: "Tux ECB effect", 128-bit version of AES-ECB with password of 'Simple Password is ample' has been applied to linux mascot known as Tux

AES, are not suitable. In terms of security, considering the inner non-iterative and nonlinear behavior, encrypt media in a much more reliable way. Moreover, some of these methods can withstand many cryptanalysis tests and real life attacks such as chosen plain-text and cipher-text attacks [12–16]. Also, successful attacks on AES-CBC has been reported and analyzed which padding oracle attack is one of these researches[17, 18].

However, chaos based methods and algorithms inherit two drawbacks from their chaotic map and employed scheme that are: 1) 1D chaotic maps (e. g. logistic) have a limited range for their chaotic behavior. Outside this range, they behave like any other regular mathematical function; 2) encryption methodology, used in these algorithms, are not quite strong to confront advanced attacks. Due to the lack of an ideal and secure media encryption/decryption process all the required requisites and promises for the mentioned necessity.

We present this research in the following order: in Section 2 we briefly review the previous studies on this issue. We discuss 1D chaotic maps and combined chaotic systems in Section 3. Section 4 proposes a polynomial coupling of 1D chaotic systems and its testing results are presented in Section 4.1. Section 5 explains our proposed algorithm for image encryption. We discuss cryptanalysis results in Section 6. Finally, we come to a conclusion in Section 7.

2. Related Study

Researchers have introduced various image encryption algorithms that are based on chaotic maps in different studies [16, 19–49]. Such algorithms mainly use one or multiple dimensional maps as a random sequence generator. These series are then utilized to make a good diffusion and confusion over output. For such crypto-systems, the secret key, independently or combined with some parts of source image, are considered to be seed [50].

Among the mentioned methods, there are techniques which take the advantage of using two chaotic map coupling as a new system [47]. Considering the chaotic sequence, this image encryption technique applies a pixel-wise permutation in order to provide more random behavior on the encrypted image. The results of this mentioned process, which takes four rounds, show good statistical properties.

Chaotic image encryption algorithm described in [48] utilizes logistic/sine combination and names it LS for short. This map is also a sequence generator where its key functions as seed. Furthermore, It can be used for decision making on multiple levels. The authors intended to permute or substitute source image regarding this sequence. Just like the previous algorithm, these operations are performed in pixel level. The short discussion that is made on the technique intends to show how strong it is; but the results themselves are inadequate to judge.

On the other hand, authors of [20] improve the same chaotic coupling technique in [48], in order to generalize the chaotic system. The generalization is resulted in a new system where any two 1D chaotic maps are combinable. The provided encryption scheme is very simple and ample enough to create confusion and diffusion on encrypted image. In this technique, Image rotation, pixel substitution and row separation/combination operations shape the encryption algorithm.

Another novel technique comes from bitplane decomposition of pixels in order to separate an image into bit order segregated sub images [51]. This algorithm that is named *DecomCrypt* has been employed in creating an image encryption crypto-system. The process of this algorithm starts with a binary Fibonacci bitplane decomposition of key and image along with a XOR operation between them. Then a scrambling operation completes the round. It furthermore runs N-iterations in which N is a variable number between 1 and 3.

Bitplane decomposition was later used in [45] with PWLCM (piecewise linear chaotic map). The algorithm uses bitplane decomposition and transformation of result into a row format, then applies confusion and diffusion based on two keys (key_1 and key_2) that were extracted from secret key. The output after N-iterations is transformed into bitplane row vector and combined to obtain encrypted image.

Another chaos based image encryption system is proposed in [46]. This system uses a 2D cat map in a modified fashion that is also an inner function on permutation layer. First layer of encryption process is diffusion layer, the output of this layer is converted to binary format and is followed by the permutation layer. At the end the binary sequence of image is converted back to integer numbers. Secret key in whole process is a seed for chaotic sequence generation.

The combination of 1D chaotic maps with a novel image encryption algorithm is the content of work done in [44]. Considering this new chaotic system, the authors discussion shows how secure a sequence generation can perform as permutation and substitution operations. Through this process, regarding the input key, a sequence generation permutes and diffuses image. In the next step, the encrypted image is produced by the linear transformation at the end of the crypto-system. All operations which is applied in the described algorithm run on color image. The statistical results along with the cryptanalytic discussion confirm its security and reliability.

Other novel methods such as combination of fuzzy cellular neural networks (FCNN) into cryptography is proposed [49, 52]. Chaotic fuzzy cellular neural network based approach presented in [52] uses a fast and elegant method. Their proposed encryption algorithm generates a chaotic signal with respect to image pixels and input keys (A, B, D, α, β) and mixes image pixels with regards to this signal. Their proposed method shows acceptable cryptanalytic results and statistical tests such as NPCR and UACI. On the other hand, M. Kalpana et al. presented another methodology utilizing drive-response-type chaotic delayed FCNNs [49]. Authors used two bounded and continuous functions, ϕ and φ, for two different systems as initial conditions. Their algorithm with fuzzy feedback MIN/MAX template shows promising statistical and cryptanalytic results.

Compressive sensing is another novel method proposed recently [53]. Zig zag path for confusion introduced by authors with initial starting index based on secret key is a novel approach for swapping image pixel values. Discrete wavelet transform, SHA-512 hash function and Memristive chaotic system are other inner functions of their encryption algorithm.

Although the previously described algorithms and schemes require a key in shape of floating point number, in real case scenarios, users attend to input the key as a string of ASCII characters. In other words, seeing the whole process from programmers perspective, the binary representation of secret key is more formal and useful. However, in none of the described methods, the string key terminology is used and accordingly they assumed that the key should be in floating point format. Moreover, on previously proposed chaotic maps, the sequence generation must be a random sequence generation process; i.e. that it is not clear whether the produced sequence is a random number generator (RNG) or not. Additionally, further tests such as NIST statistical test suite is required to make this assumption clear [54].

In order to provide more confusion and diffusion on generated keys, employing the round key generation scheme is also necessary. Accordingly, most of the existing methods use the chaotic sequence generator to make this happen.

Nevertheless, more discussion about existing algorithms are related to cases where both the inner bitwise operations are mainly XOR or its variant and where no dynamic function generation is employed. The nature of dynamic function generation is a novel idea. In this case the cryptographer gains a powerful tool for designing bitwise functions at runtime and that means more random behavior on output.

To overcome the mentioned problems in this research, we have designed a new chaotic algorithm that is capable of generating dynamic functions at runtime. It's also capable of maintaining powerful statistical properties with aid of a novel chaotic system that is made by coupling of one dimensional chaotic maps.

3. Chaotic systems

Based on their functionality, chaotic maps are categorized as one-dimensional and multi-dimensional maps. To differentiate, 1D chaotic maps perform faster in terms of speed, while multi-dimensional chaotic maps have computational overhead due to their high-dimensional structure. On the other hand, 1D maps are less complicated for implementation and yet as much useful as multi-dimensional ones. Conversely, the drawback is when the 1D maps have short range of initial and input variables while multi-dimensional maps can be extended and have a wider range[55, 56].

3.1. 1D chaotic maps

As described earlier, 1D chaotic maps, accept initial conditions such as α, β, γ, etc with an initial value X_0 as an input and perform a mathematical function like $f(X_0, \alpha, \beta, ...)$. It is also considered to be a mapping from initial conditions and some control parameters to the predetermined range. Equation 1 performs a N-time composition over f chaotic iteration:

$$f_n \circ f_{n-1} \circ \ldots \circ f_2 \circ f_1 \circ f_0. \tag{1}$$

Logistic Map, which is one of the famous 1D chaotic maps, is described in Eq. 2. From the main characteristics of this map are low computational cost and high chaotic behavior ($r \in [3.57, 4)$, $X \in (0, 1)$). Although this chaotic function has limited range of r for its chaotic behavior, it also suffers from absence of extra control parameters. Hence to overcome the mentioned problems of this and other 1D maps like *Tent* and *Sine* (equations 3 and 4)[57], many modifications and chaotic system designs are proposed [20, 44, 47, 48, 58].

$$L(X, r) = rX(1 - X). \tag{2}$$

$$S(X, \alpha) = \alpha \sin(\pi X/4). \tag{3}$$

$$T(X, u) = \begin{cases} uX/2 & X < 0.5, \\ u(1 - X)/2 & X \geq 0.5. \end{cases} \tag{4}$$

3.2. Combined chaotic systems

Combined or coupled chaotic maps are referred to as combined chaotic systems. These systems use chaotic maps in order to make a new one that is furthermore capable of handling problems, risen from the source maps. The research done in [47] proposes the utilization of *i-th* element on chaotic series as a control sequence in order to select next chaotic map for *(i+1)-th* element generation. The application is demonstrated in eq. 5. In this equation β is known to be third 1D chaotic system. It controls which chaotic system is going to be used at next iteration. . It happens based on the quantity of β in a way that if it is greater than or equal to 0.5, f_1 and otherwise it is f_2 function that will be used. Both of these functions are also 1D chaotic maps:

$$F(X_n, \beta_{n-1}) = \begin{cases} f_1(X_{n-1}) & \beta_{n-1} = 0, \\ f_2(X_{n-1}) & \beta_{n-1} = 1. \end{cases} \tag{5}$$

As an earlier version of [20], in [48], a combination of *sine* and *logistic* map is proposed. In order to obtain a new system in these settings, two specific chaotic maps are required to be precisely set and combined. More advancements on the same map is obtained in [20] by utilization of modulo operator. Addition of two chaotic maps and applying modulo operator on result is shown in eq. 6:

$$F(X_n, \gamma, \lambda) = [f_1(X_n, \gamma) + f_2(X_n, \lambda)] \mod 1. \tag{6}$$

Three examples of this map is also explained by authors that combine *sine*, *tent* and *logistic* maps. Major modifications to this system has also been applied in [58] to increase parameters and make the underlying system more compressing rather than a two-to-one mapping. The compression ability proposed by authors made the system more compatible for cryptographic applications such as hash functions. Equation 7 shows the resulting chaotic system:

$$F(X_n, w_\alpha, w_\beta, \alpha, \beta, r) = [(w_\alpha \alpha f(r, X_n)) + (w_\beta \beta f(16 - r, X_n))] \mod 1. \tag{7}$$

A multiplicative-subtraction system, which is composed of a 1D chaotic map, is proposed in [44]. A *flooring* operation is used in this system in order to obtain the chaotic part, similar to modulo function application.

4. Proposed Polynomial Chaotic System

As a new chaotic system, our polynomial coupling (eq. 8) offers a novel method, in which multiple or single chaotic maps are coupled. This coupling is applied both to expand the maps input range and to obtain higher random-like behavior.

$$\begin{aligned} X_{n+1} = F_k(X_n, \Psi, \Xi) = [G_1^1 & \\ &+ (\Psi_1 G_1^2 + \Psi_2 G_2^1) \\ &+ (\Psi_3 G_1^3 + \Psi_4 G_2^2 + \Psi_5 G_3^1) \\ &+ (\Psi_6 G_1^4 + \Psi_7 G_2^3 + \Psi_8 G_3^2 + \Psi_9 G_4^1) \\ &+ \cdots + (\Psi_j G_1^k + \Psi_{j+1} G_2^{k-1} + \cdots + \Psi_{j+k-1} G_{k-1}^1)] \mod 1, \\ & \text{where } j = \sum_{i=1}^{k-1} i \end{aligned} \tag{8}$$

Here $\Psi_1, \Psi_2, \Psi_3, \Psi_4, \ldots$ parameters control effects of G_i functions on final output. On the other hand, $\Xi_1, \Xi_2, \Xi_3, \Xi_4, \ldots$ are changing for each chaotic map. $G_1, G_2, G_3, \ldots$ are the inner functions which are the same maps with even different parameters for control conditions (i.e. different chaotic functions, $g, h, \ldots$, and diverse control parameters can be employed).

A logistic example of this system is illustrated in eq. 9. For the same X, it only uses six different logistic functions with diverse control parameters. Accordingly, the logistic map is reformulated, in its original form, as $L_1(X_n, \Psi, \Xi)$ where $\Psi = \emptyset$ and $\Xi = \{r\}$.

$$\begin{aligned} X_{n+1} = L_4(X_n, \Psi, \Xi) = [l_1^1 + (\Psi_1 l_1^2 + \Psi_2 l_2^1) & \\ &+ (\Psi_3 l_1^3 + \Psi_4 l_2^2 + \Psi_5 l_3^1) \\ &+ (\Psi_6 l_1^4 + \Psi_7 l_2^3 + \Psi_8 l_3^2 + \Psi_9 l_4^1)] \mod 1, \\ & \text{where } l_i = \Xi_i X_n (1 - X_n). \end{aligned} \tag{9}$$

4.1. The chaotic behavior of proposed system

Three different orbits of our chaotic system are displayed in fig. 2 for 50 iterations, the special case of L_4 has been used for this test. In this special case, the Ψ_i values are set to $(i + 1) \times \Psi_0$ and the Ξ values are all equal to Ξ_0 where the Ξ_0 and Ψ_0 are 2.1 and 3.57 respectively. Histogram analysis is displayed in fig. 3 for one thousand different initial conditions and random Ψ and Ξ values, which each of them is containing 50 iterations, the Ψ and Ξ values are chosen randomly. Blue line in this figure indicates the mean of all cumulative sums, at different bins.

It is possible to express chaotic behavior of any map in terms of multiple analysis that are called cobweb plot and the largest lyapunov exponent. Additionally for an extra randomness testing, the NIST random number test suite is quite useful [54].

Figure 4 shows cobweb plot of the L_4 for our proposed chaotic system with various Ψ values. We can see in the figure that the folding effect happens for the higher values of equally changing Ψ's and it has illustrated in blue. We

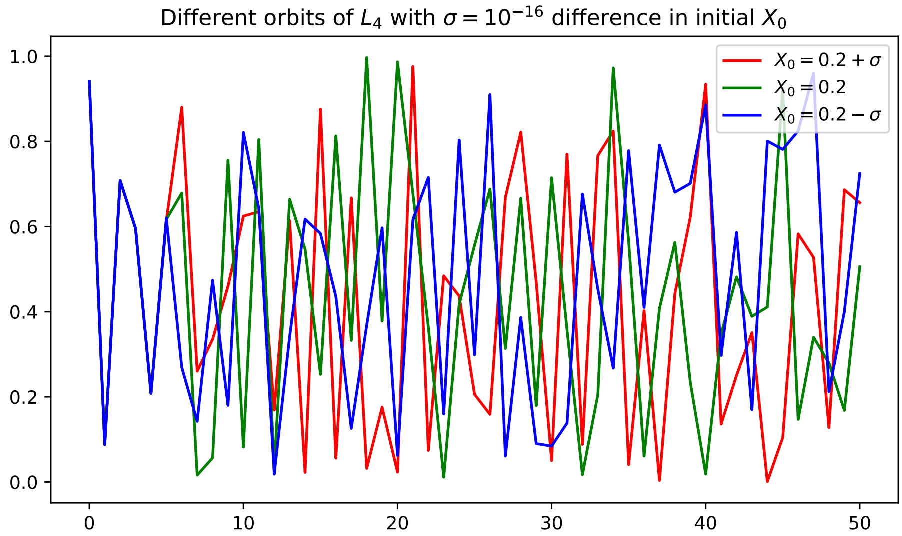

Figure 2: Three different orbits of our proposed system with variance of $\sigma = 10^{-16}$ on initial condition X_0

call this controllable effect the *Butterfly Folding*. Respective values for Ξ_i are 3.8 and logistic map is the seed map in all inner l functions of L_4.

Largest lyapunov exponent (LLE) is another useful randomness measure on chaotic series. The chaotic behavior of a particular experiment is indicated by its LLE being larger than zero. However, for the systems with LLE values under or equal to zero the behavior is considered to be non-chaotic [59]. We have applied this test to L_3, L_4 and L_5. Figure 5 shows this plot within the range of [1, 51] for Ψ_0 and [4,10] for Ξ_0 in which 10 points has been examined. Other Ψ and Ξ values has been set by $\Psi_i = (i + 1) \times (\Psi_0 + 0.001)$ and $\Xi_i = \Xi_0$ respectively. This assumption and special case of L_i functions made the LLE graph drawing possible in a 3D space that without it presenting the data would be impossible in a larger dimensionality. Lyapunov exponent values are computed by the method proposed in [60] and it can be seen from figure 5 that all functions show chaotic behavior in the mentioned domain. Respective mean and standard deviation values of each sub figure are noted in graph for clarity. Compared to TCL map proposed in [61], our approach has higher LLE mean value which their study shows approximately 0.6. For our system, mean LLE value for $L3$, L_4 and L_5 is 0.6447, 0.6362 and 0.6247 respectively.

There is also the NIST random number test suite (for cryptographic applications) that has ample tests to show the performance of a random number generator. In case of a chaotic map or system it can be considered as a Random Number Generator (RNG), if it is considered to have enough random behavior. Furthermore, we have simply multiplied the output of tested maps to 1024 and quantized values in order to map all real values between 0 and 1 to the byte values (containing 0 and 1's). Thus the output of tested maps has changed to be integer values between 0 to 1023. After that, we converted the result to bits (where each sample contains 10 bits). We have fed our generated bit stream to NIST random number test suite and have presented the associated results in table 1. This table contains a comparison between L_4 of our proposed system and the logistic, tent and sine maps. The control parameter range for any map except our system is set to its chaotic range in which the map shows chaotic behavior. For our proposed system the sequences are generated by randomly picking Ψ and Ξ values and for each picked value 100 iteration is performed. NIST random number test suite used for this examine is version 2.1.2 and the results are obtained from a linux mint operating system.

4.2. The hardware implementability of proposed system

For obtaining a high random-like system, taking the advantage of a single chaotic map is very beneficial. One of these advantages is on hardware implementation of the system when the desired system can be implemented in a parallel way. Figure 6 shows a blockwise hardware implementation example. In this example design, combinator

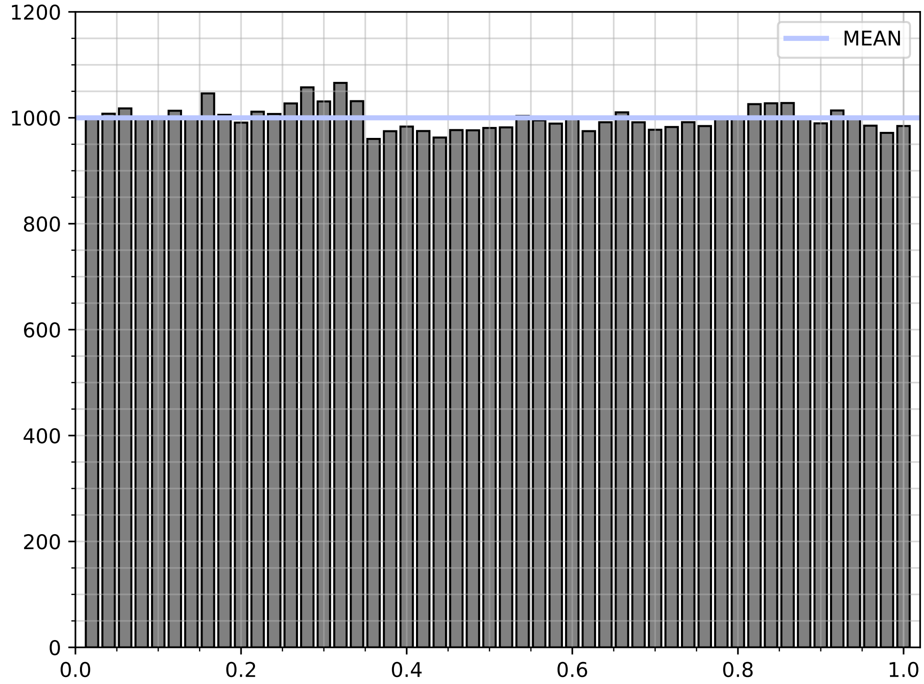

Figure 3: Histogram of our proposed system

layer gets the output of previous layer and combines it according to eq. 9, $\times$ block multiplies two inputs and the $C2$ block outputs the negative value of respective input value. This hardware implementation example shows how a L_4 chaotic system can be implemented in a parallel fashion.

5. Proposed image encryption scheme

The chaotic system that is defined in sec.4 is utilized along with multiple novel confusion and diffusion functions in order to result in good cryptographic properties. Figure 7 shows flowchart of our proposed algorithm. In our setup, six inner functions (i.e. *GenRK*, *GenXMap*, *GenPMap*, *GenSMap*, *DFLGen*, *Exchanger*) act in a binded cooperation to make the final encrypted output, shown as *Encrypted Image*. Here, secret key is assumed to be a string that is containing multiple characters in ASCII format with arbitrary size. In this scheme, an image and a secret key are the inputs to start data flow. After N-iterations, this process ends up with producing an encrypted image from the inputs. The same scheme with minor modifications is employed for decryption process. Below, we have described the workload of each inner function.

GenRK. takes secret key as input and outputs (1×20) round-key (RK) matrix, Ψ and Ξ which respectively are in shape of 1×9 and 1×4. To obtain these values from secret-key, first the key is converted to [0,1] range and equal or less than 9 or 4 number of zeros are appended to make it in appropriate format for the rest of algorithm (9 for Ψ and 4 for Ξ). XOR operation is applied to first axis of obtained matrix to obtain 9 or 4 numbers and the obtained numbers are combined with rest of matrix with same operation. For Ξ needs to be in range of [3.57,4] to obtain chaotic behavior from logistic map. Thus, each value of Ξ_i in Ξ is divided by 5 and added to 3.8. In order to have an avalanche effect between previous roundkeys and current round key that is to be produced, each RK from previous iteration is added to 20'th iteration of L_4 map with obtained Ψ and Ξ values and is corrected to [0,1] range afterwards.

GenXMap. RK, Ψ, Ξ and size of input image are considered as input and the output is XOR-Map, which is denoted as XMap. This map is later used by *Exchanger* and *GenPMap* functions. Algorithm 2 shows how the inputs are combined, in order to iterate $M \times N$ times over L_4 function.

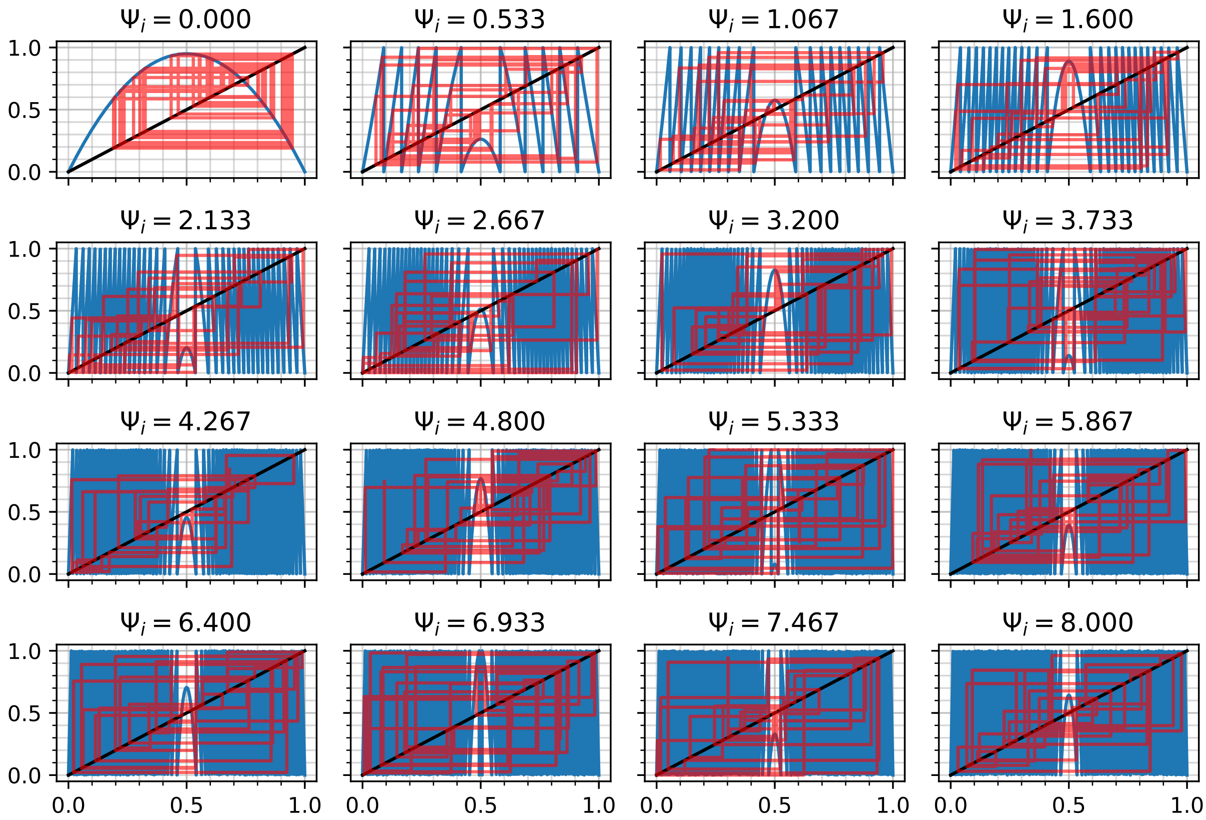

Figure 4: *Butterfly Folding* effect and Cobweb plot for l_4 with various Ψ values

Algorithm 1 GenRK

Require: $SecretKey$ (1×arbitrary matrix)
  SecretKey= SecretKey / 256
  temp= Append $9 - mod(SecretKey, 9)$ number of 0's to SecretKey
  temp= Convert temp from 1×K to floor(K/9)×9
  Ψ= ⊕ of temp on first axis
  Ψ= Ψ ⊕ (with any cell in temp)
  temp= Append $4 - mod(SecretKey, 4)$ number of 0's to SecretKey
  temp= Convert temp from 1×K to floor(K/4)×4
  Ξ= ⊕ of temp on first axis
  Ξ= Ξ ⊕ (with any cell in temp)
  Ξ= 3.8 + Ξ/5
  for any c in $SecretKey$ **do**
    RK= RK + $iter_{L_4}(\Psi, \Xi, c)_{20times}$
  end for
  RK= mod(RK,1)
  return RK, Ψ, Ξ

GenPMap. sorts XMap and assigns a number for each item after sorting. The new matrix, which is the index of sorted elements, shows how a permutation must be acquired according to this setting. Obtaining inverse operation of

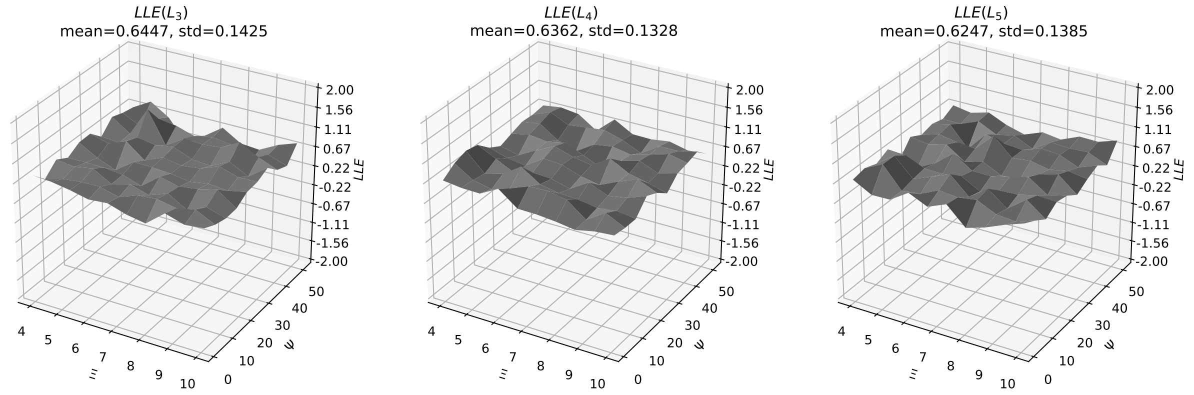

Figure 5: LLE for L_3, L_4 and L_5

Table 1: NIST statistical random number test results for cryptographic applications: test has been applied for 10^6 sequences of 100-bits generated from Logistic, Sine, Tent and L_4, minimum pass rate for each test is approximately 96

Statistical test	p-value				Proportion			
	L_4	Logistic	Sine	Tent	L_4	Logistic	Sine	Tent
Frequency	0.739918	0.000000	0.000000	0.000000	**99**	*86*	*31*	*36*
Block Frequency	0.574903	0.000000	0.012650	0.000000	**100**	**100**	**99**	*32*
Cumulative Sums	0.779188	0.000000	0.000000	0.000000	**100**	*87*	*36*	*35*
Runs	0.366918	0.000000	0.000000	0.000000	**99**	*87*	*50*	*21*
Longest Run	0.964295	0.000184	0.000216	0.000000	**99**	*92*	**97**	*51*
Rank	0.437274	0.000082	0.102526	0.006196	**99**	**99**	**99**	**99**
FFT	0.007694	0.000000	0.779188	0.000000	**99**	*78*	**99**	*88*
Non Overlapping Template	0.275709	0.042808	0.000043	0.000000	**100**	**100**	**100**	*33*
Overlapping Template	0.035174	0.437274	0.595549	0.000015	**100**	**100**	**98**	*81*
Serial	0.006196	0.000000	0.000000	0.000000	**100**	*0*	*0*	*0*
Linear Complexity	0.085587	0.678686	0.319084	0.191687	**100**	**96**	**96**	*91*
Approximate entropy	0.834308	0.000000	0.000000	0.000000	**99**	*0*	*2*	*6*
Universal	0.419021	0.000000	0.000000	0.000000	**99**	*0*	*0*	*0*

such matrix is ample to reverse the permutation process. This function is described in alg. 3.

GenSMap. substitution matrix for all pixels in one iteration is the output of 255 iterations on L_4. This task is assigned to GenSMap that is described in algorithm 4.

Function List. is atomic functions such as basic bitwise operations that form the initial function list. This function list is composed of six tuple elements and in each tuple, operations are dual based on each other. Equation 10 shows this list. The mentioned list can be extended in order to make more confusion and diffusion on output for other setups. The list includes: Bitwise rotation to left and right (rol, ror), XOR operation, complementation (not), XNOR operation and the no-operation (noOp: do nothing). For encryption process the operations and for decryption process the duals are used.

$$FunctionList = \{(ror, rol), (rol, ror), (xor, xor), (not, not), (noOp, noOp), (xnor, xnor)\}. \tag{10}$$

DFLGen. dynamically shuffles function list and outputs a new function list of operations. These operations are later used by *Exchanger* in a cascading form to make the final output. Algorithm 5 explains *DFLGen*.

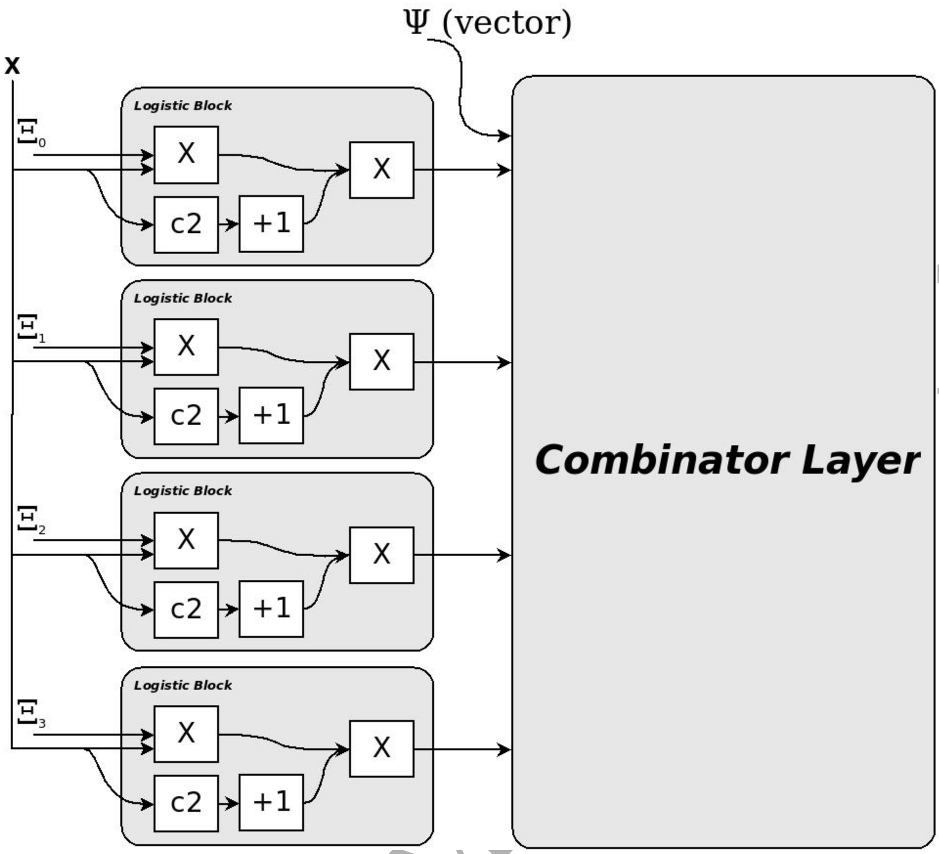

Figure 6: Parallel hardware implementation example of proposed chaotic system

---

Algorithm 2 GenXMap

Require: rk (denotes a single roundkey), Ψ, Ξ, $M \times N$ (image matrix size)

  $XMap$= $iter_{L_4}(\Psi, \Xi, rk)_{M\times N}$

  $XMap$= $floor(XMap \times 255)$

  return $XMap$

Exchanger. gets inputs from previous functions, for the last iteration it computes the final encrypted image, otherwise it computes its own input for the next round. This input takes place of *SourceImage* as an intermediate value. Substitution, permutation, XMap and function list influx computation are other steps of this function. Algorithm 6 explains the process in more details.

The process of image encryption starts with *GenRK*. After generation of round-keys, which are noted as *rk* in a *RK* matrix, the $GenXMap$ generates a $XMap$ based on $RK0$. This map is as same size as the image and is also fed to $GenPMap$ function to compute permutation matrix denoted as $PMap$. Simultaneously, in a multi-threaded fashion, $GenSMap$ and $DFLGen$ form substitution matrix and dynamic function list, respectively, according to $RK0$. At the next step, by using $PMap$, $Exchanger$ permutes source image and then substitutes each pixel according to $SMap$. Further, it applies XOR operation between the intermediate result of previous two steps and $XMap$. At the end, each function in dynamic function list, from $DFLGen$, is operated at each pixel, in a cascading mode over intermediate image. The whole process repeats for each *rk* in *RK* matrix; i.e. it takes 20 inner iterations. For the last iteration, the result is the encrypted image otherwise the result is the intermediate image. As it's mentioned earlier, each iteration is twenty rounds of the whole process.

The decryption process is same as encryption with minor modifications. Encrypted image is an input to the

Algorithm 3 GenPMap

Require: rk, Ψ, Ξ
  $XMap$= $GenXMap(rk, \Psi, \Xi, M \times N)$
  $PMap$= a number according to sorted position of each element in $XMap$
  return $PMap$

Algorithm 4 GenSMap

Require: rk, Ψ, Ξ
  $SMap$= $iter_{L_4}(\Psi, \Xi, rk)_{256times}$
  $SMap$= a number according to sorted position of each element in $SMap$
  return $SMap$

Algorithm 5 DFLGen

Require: rk, Ψ, Ξ
  $temp$= $iter_{L_4}(\Psi, \Xi, rk)_{6times}$
  $FunctionList$= Sort initial Function list according to $temp$
  return $FunctionList$

Algorithm 6 Exchanger

Require: $SourceImage$, $XMap$, $SMap$, $PMap$, $FunctionList$, RK
  $EncodedImage$= Permute $SourceImage$ according to $PMap$
  $EncodedImage$= Substitute any $pixel$ according to $SMap$
  $EncodedImage$= $EncodedImage \oplus XMap$
  for any $function$ in $FunctionList$ **do**
    for any rk in RK **do**
      $EncodedImage$= $function(EncodedImage, rk)$
    end for
  end for
  return $EncodedImage$

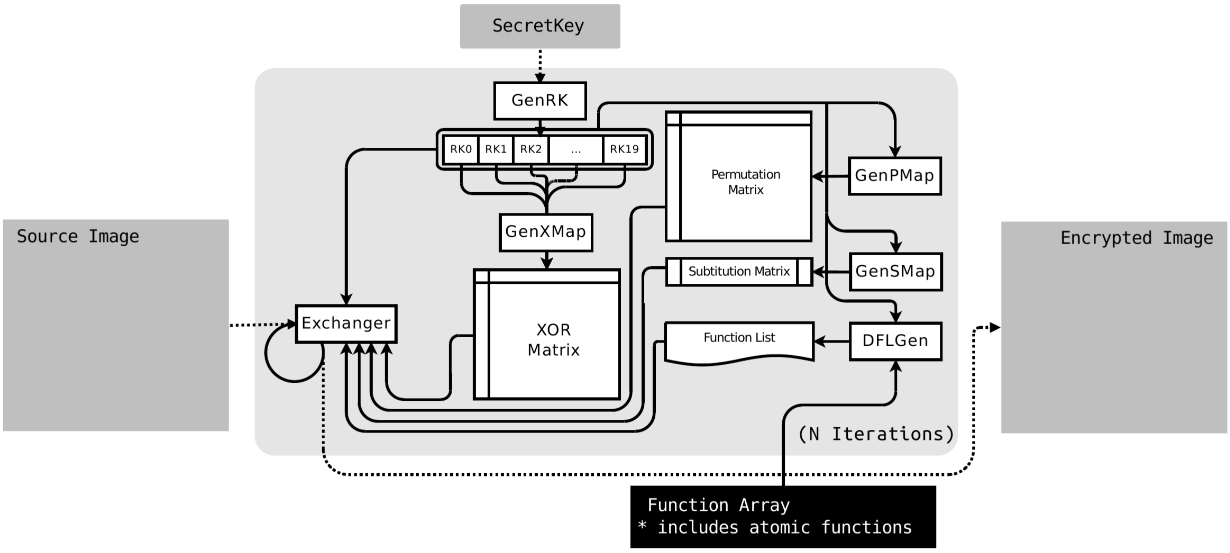

Figure 7: Proposed image encryption algorithm

encryption process along with the secret key, *GenRK* function generates *rk* values in *RK* with respect to provided secret key, *GenXMap*, *GenPMap* and *GenSMap* generate the respective XOR, permutation and substitution maps. The dynamic function list used here is there reverse form of eq. 10 in which the second function in each tuple is used. *Exchanger* permutes the encrypted image and and substitutes each pixel according to *SMap*. Intermediate values of previous iteration and current are XORed and at end of each iteration, each function of dynamic function list is applied to each pixel in a cascading mode. As number of iterations the encryption process took, the decryption process takes same to decrypt the encrypted image.

6. Cryptanalysis of the proposed algorithm

An example of our algorithm is provided in Figure 8 where the secret key is ”Simple Password is ample”. As it is clear from fig. 8 the TuxECB effect is not present anymore and has been vanished during the first analysis of our algorithm.

For a typical encryption algorithm, other performance metrics are defined in terms of statistical analysis and are known as cryptanalysis attacks. All of these metrics are covered in subsections 6.1 and 6.2.

6.1. Statistical analysis

Histogram Analysis. For an encrypted image, it is necessary to have a uniform distribution over its pixel values on both axes. This property can be simply extracted from histogram of image and related analysis, such as chi-square test. Chi-square test shows whether a distribution assumption (uniform distribution in our case) is close to observation (encrypted image frequency distribution over possible pixel values) or not. Figure 9 shows ”Lena” image, its histogram and respective plots for encrypted image. Equation 11 introduces chi-square. The χ^2 values obtained for three different pictures are listed in table 2. Encryption quality is another measure which is defined by deviation between the original and encrypted image on each byte level, encryption quality is a good metric to show statistical fitness of algorithm but the χ^2 covers this metric too as it is clearly seen from the χ^2 definition.

$$\chi^2 = \sum_{i=0}^{N-1} \frac{(o_i - e_i)^2}{e_i}. \tag{11}$$

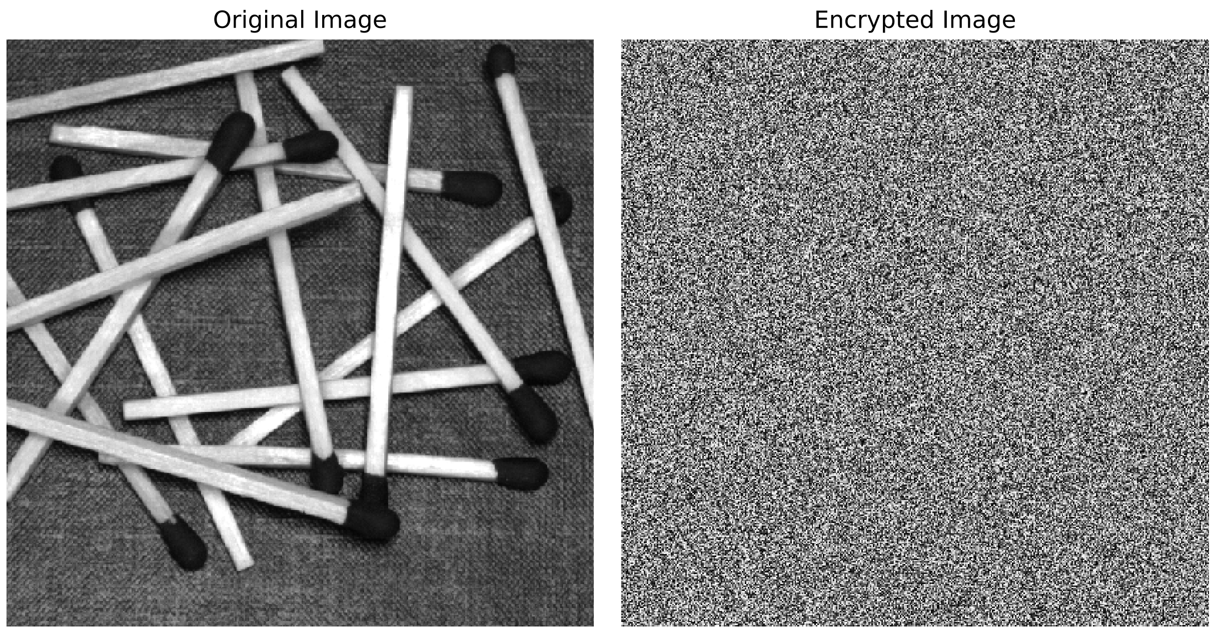

Figure 8: Source Image (Sticks) and its respective Encrypted Image with password set to "Simple Password is ample"

Table 2: Chi-square test of histograms for three different images of our method with two different passwords: P_1="Simplicity of password" and P_2="S0mEH1R5P&ˆ7.asW!"

Encrypted image	Lena	Sticks	Peppers
Chi-Square test P_1	262.054	259.324	258.956
Chi-Square test P_2	257.672	258.493	261.170

In this equation, N denotes number of color levels (256 in our case), o_i is the observations (frequency of each color level i) and e_i is the uniform distribution value. This value is 256 for all grayscale images of size 256×256. If we assume that the encrypted image should have a uniform distribution. The result of such case scenario for an encrypted image should be as low as possible. The higher values of chi-square imply how much a typical observation is far from hypothesis that we made. In our case, the result was near 260, meaning for all 256 color levels, we have 260 error in observations, and for each color level the value is close to 1 on average. In other words, any color level is typically, roughly, one value higher or lower than the hypothesis.

Number of Pixels Change Rate (NPCR). defines number of pixels that are not identical in two images (eq. 12). Higher values for NPCR denote higher resistance against brute force attacks on key space. This metric is applied to different encrypted images that are obtained from using same method for different passwords. For applying this test, we have generated 2,000 different passwords with minimal distance (one character was shuffled at each time). Additionally, we tested our algorithm against other methods and results of the described test that indicates key sensitivity and resistance to brute-force attacks are demonstrated in Table 3 for various case scenarios.

$$NPCR = \frac{1}{M \times N \times K} \times \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{k=1}^{K} Q(I_{m,n,k}, I'_{m,n,k}) \times 100(\%). \qquad (12)$$

M, N and K in Eq. 12 are image matrix size (k=3 for RGB image and K=1 for gray scale). Also, I and I' are the two images that need to be compared by Q function. Whenever both pixels in same location are equal Q function equals to zero, and is one in other cases.

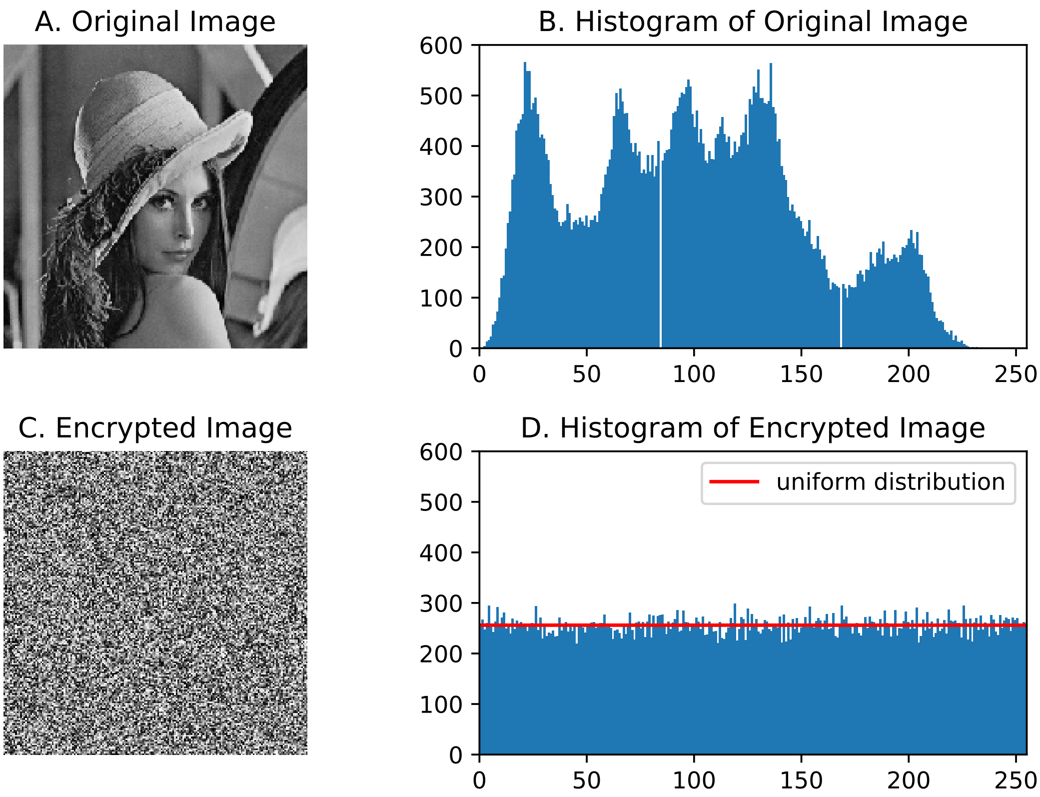

Figure 9: Source Image (Lena) and its respective Encrypted Image along with histograms of both, password is set to ”Simple Password is ample”

Unified Averaged Changed Intensity (UACI). is defined as average intensity change, between two images at same position. This metric determines distance between two pixels while the NPCR is only a metric, for unequal values. Equation 13 defines the UACI.

$$UACI = \frac{1}{M \times N \times K \times L} \times \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{k=1}^{K} |I_{m,n,k} - I'_{m,n,k}| \times 100(\%). \tag{13}$$

In the above equation, L is the highest color level. M, N and K are same as eq. 12.

Information Entropy. is another powerful measurable property that is defined as number of required bits to represent data without loss (eq. 14).

$$H(m) = \sum_{i=0}^{2^{n-1}} P(m_i) log_2 \frac{1}{P(m_i)}. \tag{14}$$

In the above equation, $P(m_i)$ is the probability of symbol m_i. For our algorithm, we have averaged the obtained entropy from 100 computed images. The resulting value for this test is equal to 7.999334 (the maximum entropy achievable is 8). Table 3 reports the information entropy for different images. Moreover, table 5 presents UACI, NPCR and Information Entropy values for the proposed algorithm and state of the art algorithms.

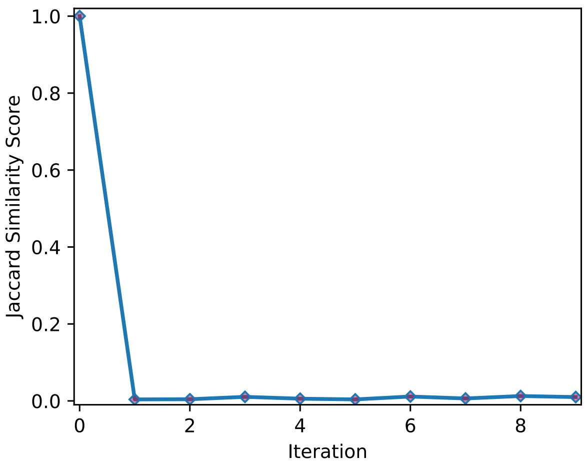

Figure 10: Jaccard similarity score between source and encrypted image for nine iterations

Shannon Block Entropy Randomness Test. is proposed by [62] and uses multiple random windows with same size over encrypted image to compute entropy.

$$\overline{H}_k = \sum_{i=1}^{K} \frac{H(Y_i)}{K}. \tag{15}$$

The original test proposed by authors shows that randomness across an image must be also tested locally and according to them, if any trial with entropy under 7.1627674499 results in a fail. The mean and standard deviation values computed from multiple trials is useful to test the encryption algorithm and compare it to others. Equation 15 shows block entropy mean value for K windows. Table 6 shows test results of our proposed algorithm compared to other methods. For this test K is set to 100 and *USC-SIPI Miscellaneous* image dataset [63] is used. Mean values are average block entropy for whole dataset and *std* denotes the standard deviation while number of fails are also presented. The italic typed values show the fails and bold ones presents the highest score for this test.

Table 3: Entropy, NPCR and UACI for key sensitivity

Image	NPCR (%)	UACI (%)	Entropy
Lena (256×256) RGB	99.7024	33.5249	7.9984
Lena (512×512) GS	99.6912	33.5098	7.9975
Barbara (256×256) GS	99.7144	33.5137	7.9981
Barbara (512×512) GS	99.7695	33.4979	7.9985

Jaccard Similarity Score. is known to be the proportion of elements that intersect to all of elements in both samples. For two images, this score is defined as the rate number of pixels, on both images that are in same position to all of pixels (eq 16). In order to determine the mentioned score for our algorithm, we executed a test with nine iterations, starting from zero and then we calculated the jaccard similarity score. The results are presented in fig. 10.

$$JaccardSimilarityScore = \frac{|I \cap I'|}{|I \cup I'|}. \tag{16}$$

In above equation, result of $\cap$ is related to the pixels that are in same position with equal value. Furthermore, $\cup$ denotes union of all pixels.

Correlation Coefficient. on horizontal, vertical and diagonal axes shows relationship between two adjacent values. These two values (x, y) are adjacent pixels in source or encrypted image and their adjacency is defined in of horizontal, vertical or diagonal ways. Equation 17 calculates the correlation coefficient between two matrices. Moreover, the calculated values of this parameter for different images related to three axes are compared to state of the art algorithms in Table 4. Figure 11 illustrates the same analysis for source and encrypted images.

$$C_{x,y} = \frac{E[(x-\mu_x)(y-\mu_y)]}{\sigma_x\sigma_y}. \tag{17}$$

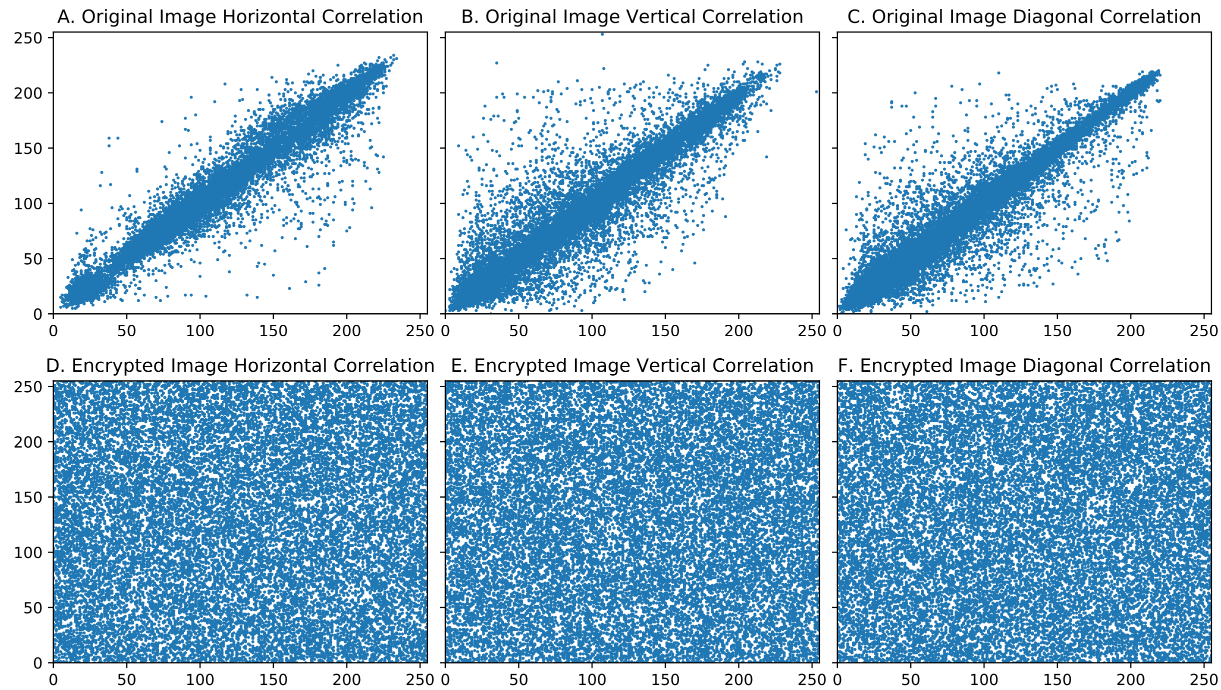

Figure 11: Correlation plot of source and encrypted image in diagonal, horizontal and vertical adjacency axis

6.2. *Known Cryptanalytic Attacks*

Triune cryptographic parts of a cryptanalytic process involves: Source image, Encrypted image and secret key analysis. After the first part of the analysis one may trust the security adequacy of the key against the attacks, but as the operation performance continues over the source image, this breaks because only statistical features are put into mind. Thus, there is a meaningful relation between key and image that can be extracted according to the computed output. Second part assumes that the encrypted image contains some semantic relation over key and the source image. The applied assumption causes the relation not to be random enough. Regarding the decrypted image, some minor changes in assumption can lead to a meaningful relation. At the end, the key will be gone through some tests like brute-force, minor-changes and etc. At this phase, the main goal is to find out the algorithm performance on fixed points, in which the process is the same interdependent of the nature of inputs and outputs. In the following subsections, we focus on known attacks.

Key Space. is a very important property of every cipher. In the case of short key spaces, the result is a reduction in trials; The ones that a typical attackers needs to search blindly. In our algorithm, the key space that is in string format, starts from one to any length. Moreover, each index is also a character ranging from 0 to 255. This strong start makes

Figure 12: Manipulation Test: A) Some parts of source image is masked by zero values, B) White noise has been added to image, C) White noise has been added to Encrypted Image, D) Some parts of Encrypted image is masked by zero values, E,F) Encrypted images of A and B, G,H) Decrypted images of C and D

the user able to select its own key, based on the way one wants the algorithm to act. If a more secure encryption is required, the length of string should be larger. The input $SecretKey$ and its arbitrary length, makes a key space of $K_{space} = 256^{SK_L}$, where SK_L is length of $SecretKey$.

Key Sensitivity. is known to be the sensitivity that an algorithm shows for minor changes on $SecretKey$ of the encrypted image . In the case that the number of made diffusion and confusion is not enough, then the attacker can find out how far/close is one's hypothesis on $SecretKey'$ from the $SecretKey$. This nonlinear behavior helps to improve underlying algorithm to withstand many related attacks as Differential, Impossible Differential attack, Meet In The Middle, Bi-clique cryptanalysis and many others. Statistical analysis discussed in previous subsection, specially in NPCR and UACI, confirms that there is not any linear relation between the key and respective encrypted image.

Chosen Source Image attack. if we assume for an image such as I, that some part of source image is replaced with the lowest possible intensity value, therefore, on zero values, the attacker might get the secret key out of encryption process. Figure 12 shows an example of discussed attack and the resulted cipher image based on this assumption. Four scenarios are shown in this figure. Scenario A is masking attack where large number of pixels in source image are masked and after encryption the result is analyzed in order to obtain the secret key. Scenario B shows effect of white noise to image and encrypted image which also yields no useful cryptanalytic information about the secret key. The reverse form of these scenarios are shown in C and D where the encrypted image is masked in C and white noise added to it, is decrypted. White noise rate is 0.2 and is added randomly to whole image. We cannot apply this attack on our algorithm because SMap and PMap effects on the source image prevent such a hypothesis to become real. On the other hand, randomly and sometimes wisely chosen source image manipulations can lead to good cryptanalytic results. For example, a linear algorithm which is acting the same on known values, leads the attacker to a possible clue on the candidate $SecretKey$s which make these inner operations. Due to high confusion and diffusion of proposed algorithm, mentioned methodology is not applicable.

6.3. Performance and Speed

Performance of an encryption algorithm in terms of speed and cpu usage is another metric to compare different approaches while it has two sides. If an algorithm has higher speed and low key space, it will yield to a unsafe and fast algorithm in which the attacker can try keys on a brute force attack. Also, a slow algorithm with large key space can also yield to better security but higher power consumption and lower encryption data rate.

Table 7 presents a comparison between presented study and related state of the art algorithms. Results of our proposed algorithm has been obtained by a Linux Mint operating system on a Intel(R) Core(TM) i7 CPU (7700HQ 3.80 GHz) system with 16GB of RAM memory. The results for other methods are reported in [64–66] and directly included in table 7. As it is clearly seen from table 7, multiple systems has been utilized for speed analysis which yields in an amigiuity for making a good comparision between different methods. To overcome this problem, we use *Encryption Throughput* (ET) and *Number of Cycles Per Byte* (NCPB) from [67] that are presented in equations 18 and 19 respectively.

$$ET = \frac{Image_{Size}(Byte)}{Encryption_{Time}(seconds)} \tag{18}$$

$$NCPB = \frac{CPU\ Speed(Hertz)}{ET} \tag{19}$$

7. Conclusion

In present research, we propose a new chaotic system based on polynomial combination of 1D chaotic maps. The analytic and numerical tests confirm that this novel system is suitable for cryptographic applications specially for image encryption. The random sequence generation ability of this system has also utilized a novel image encryption algorithm. The algorithm contributes extensively on key generation, key presentation and also dynamic generation of inner functions. Statistical tests, cryptanalytic discussions and various standard simulations has tested this novel algorithm; Hence its validity has proved. Furthermore, the results of this new crypto-system on correlation tests are also compared with the state of the art algorithms and its superiority is confirmed. The high resistance to cryptographic attacks is another aspect of this algorithm which was tested. The hardware implementation o this new system can be investigated, as the future and follow up steps, for hand-held devices and cryptographic use cases.

Table 4: Correlation Analysis of Proposed algorithm

Algorithm	Image	Image Size	H Image	V Image	D Image	H Encrypted	V Encrypted	D Encrypted
Proposed Algorithm	Lena	256×256 GS	0.97187	0.98539	0.96689	**-0.00222**	**0.00137**	0.00291
El Assad et al. [46]	Lena	256×256 GS	0.97165	0.98730	0.95440	0.00312	-0.00317	-0.00310
Song et al. [42]	Lena	256×256 GS	0.96592	0.94658	0.92305	-0.00550	0.00411	**0.00021**
W. Zhang et al. [43]	Lena	256×256 GS	0.97173	0.98478	0.96869	-0.00422	0.00055	0.00366
Proposed Algorithm	Lena	512×512 C	0.99810	0.99242	0.99541	0.00271	0.00136	**0.00114**
El Assad et al. [46]	Lena	512×512 C	0.99233	0.99694	0.98712	**-0.00158**	0.00159	-0.00147
Wong et al. [40]	Lena	512×512 C	0.09751	0.98892	0.96704	0.00681	**0.00078**	0.00323
Proposed Algorithm	Barbara	512×512 GS	0.88642	0.94321	0.87651	**-0.00139**	0.00435	**-0.00109**
El Assad et al. [46]	Barbara	512×512 GS	0.89538	0.95887	0.88304	0.00155	0.00163	0.00148
Chen et al. [23]	Barbara	512×512 GS	0.91765	0.95415	0.90205	0.01183	**0.00016**	0.01480
X. Zhang et al. [39]	Barbara	512×512 GS	0.86061	0.95982	0.87741	0.00824	0.00036	0.00128

Table 5: The proposed algorithm in terms of UACI, NCPR and Information Entropy compared to other state of the art algorithms

Algorithm	UACI (%)	NCPR (%)	Information Entropy
Proposed Algorithm	**33.6751**	99.6191	**7.999334**
Zhang et al. [43]	33.4988	99.6155	7.999325
Zhou et al. [51]	33.4854	99.1891	7.999249
Zhu et al. [68]	33.467	99.6132	7.999302
Zhang et al. [69]	33.5625	**99.6254**	7.999274
Xu et al. [45]	33.4934	99.6167	7.997200
El Assad et al. [46]	33.4600	99.6095	Not Reported

Table 6: The Block Entropy Test for Encrypted Images of *USC-SIPI Miscellaneous* Image Dataset using 100 image blocks at size of 16-by-16

Algorithm	Mean	std	# of fails
Original Image	*4.9627633737*	1.7133301636	*28*
Proposed Algorithm	**7.1783666771**	**0.0072293490**	**0**
BlowFish [70]	*6.6975405565*	1.1895800576	*12*
AES [2]	*6.7488211774*	1.0726741341	*10*
TwoFish [71]	*6.7511953801*	1.0564646259	*11*
3DCat [72]	7.1718531782	0.0093951696	*2*
Sudoku [73]	7.1772687030	0.0075774206	**0**

Table 7: Performance analysis and comparison

Method	Execution Time (ms)			ET (MBps)	NCPB
	256×256	512×512	1024×1024		
Wu et al. [74]	7641	34768	151709	0.1756	17084.2824
Mohamed [75]	189	758	3097	7.8671	381.3349
Liao et al. [76]	569	2251	8986	2.6575	1128.8805
Amina & Mohamed [64]	48	139	481	41.4371	72.3988
Yavuz et al. [65]	109	391	1640	3.4250	668
Y. Wang et al. [77]	7.79	31.16	124.64	24.06	122.85
W. Zhang et al. [78]	7.5	30	120	25	122.07
A. Akhshani et al. [79]	14.4	57.6	230.4	13.02	194.83
K.-W. Wong et al. [80]	15.59	62.37	249.48	12.03	245.7
M. Farajallah et al. *V1*	2.04	8.08	31.85	93.817	31.51
M. Farajallah et al. *V2-8 bit*	**1.38**	**5.42**	**21.17**	**140.776**	**21**
M. Farajallah et al. *V2-32 bit*[66]	4.15	16.56	66.12	45.347	65.19
Proposed Algorithm	102	281	717	23.1769	163.9563

Acknowledgements

This work is supported by University of Tabriz, grant number `S/819`.

References

[1] W. Stallings, Network security essentials : applications and standards, Prentice Hall, ISBN 0130160938, 2011.

[2] J. Daemen, V. Rijmen, K. U. Leuven, AES Proposal : Rijndael, Complexity (1999) 1–45URL `http://ftp.csci.csusb.edu/ykarant/courses/w2005/csci531/papers/Rijndael.pdf`.

[3] A. Bogdanov, D. Khovratovich, C. Rechberger, Biclique cryptanalysis of the full AES, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7073 LNCS, ISBN 9783642253843, ISSN 03029743, 344–371, 2011.

[4] J. Wu, X. Liao, B. Yang, Cryptanalysis and enhancements of image encryption based on three-dimensional bit matrix permutation, Signal Processing 142 (2018) 292–300.

[5] C. Li, B. Feng, J. Lü, Cryptanalysis of a chaotic image encryption algorithm based on information entropy, arXiv preprint arXiv:1803.10024 .

[6] R. Rhouma, S. Belghith, Cryptanalysis of a new image encryption algorithm based on hyper-chaos, Physics Letters A 372 (38) (2008) 5973–5978.

[7] R. Rhouma, E. Solak, S. Belghith, Cryptanalysis of a new substitution–diffusion based image cipher, Communications in Nonlinear Science and Numerical Simulation 15 (7) (2010) 1887–1892.

[8] C. Cokal, E. Solak, Cryptanalysis of a chaos-based image encryption algorithm, Physics Letters A 373 (15) (2009) 1357–1360.

[9] D. Xiao, X. Liao, P. Wei, Analysis and improvement of a chaos-based image encryption algorithm, Chaos, Solitons & Fractals 40 (5) (2009) 2191–2199.

[10] G. Alvarez, S. Li, Some basic cryptographic requirements for chaos-based cryptosystems, International Journal of Bifurcation and Chaos 16 (08) (2006) 2129–2151.

[11] S. Lian, J. Sun, Z. Wang, Security analysis of a chaos-based image encryption algorithm, Physica A: Statistical Mechanics and its Applications 351 (2-4) (2005) 645–661.

[12] H. Guojie, F. Zhengjin, M. Ruiling, Chosen ciphertext attack on chaos communication based on chaotic synchronization, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 50 (2) (2003) 275–279, ISSN 10577122.

[13] Y. Zhang, D. Xiao, Cryptanalysis of S-box-only chaotic image ciphers against chosen plaintext attack, Nonlinear Dynamics 72 (4) (2013) 751–756, ISSN 0924090X.

[14] R. Rhouma, S. Belghith, Cryptanalysis of a chaos-based cryptosystem on DSP, Communications in Nonlinear Science and Numerical Simulation 16 (2) (2011) 876–884, ISSN 10075704.

[15] X. Wang, L. Liu, Cryptanalysis of a parallel sub-image encryption method with high-dimensional chaos, Nonlinear Dynamics 73 (1-2) (2013) 795–800, ISSN 0924090X.

[16] C. Çokal, E. Solak, Cryptanalysis of a chaos-based image encryption algorithm, Physics Letters, Section A: General, Atomic and Solid State Physics 373 (15) (2009) 1357–1360, ISSN 03759601.

[17] S. Vaudenay, Security Flaws Induced by CBC PaddingApplications to SSL, IPSEC, WTLS..., in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 534–545, 2002.

[18] J. Manger, A chosen ciphertext attack on RSA optimal asymmetric encryption padding (OAEP) as standardized in PKCS# 1 v2. 0, in: Annual International Cryptology Conference, Springer, 230–238, 2001.

[19] X. Zhang, W. Chen, A new chaotic algorithm for image encryption, in: ICALIP 2008 - 2008 International Conference on Audio, Language and Image Processing, Proceedings, ISBN 9781424417230, ISSN 09600779, 889–892, 2008.

[20] Y. Zhou, L. Bao, C. L. P. Chen, A new 1D chaotic system for image encryption, Signal Processing 97 (2014) 172–182, ISSN 01651684.

[21] X. Huang, Image encryption algorithm using chaotic Chebyshev generator, Nonlinear Dynamics 67 (4) (2012) 2411–2417, ISSN 0924090X.

[22] N. K. Pareek, V. Patidar, K. K. Sud, Image encryption using chaotic logistic map, Image and Vision Computing 24 (9) (2006) 926–934, ISSN 02628856.

[23] G. Chen, Y. Mao, C. K. Chui, A symmetric image encryption scheme based on 3D chaotic cat maps, Chaos, Solitons and Fractals 21 (3) (2004) 749–761, ISSN 09600779.

[24] D. Zhang, F. Zhang, Chaotic encryption and decryption of JPEG image, Optik 125 (2) (2014) 717–720, ISSN 00304026.

[25] Z. Tang, X. Zhang, W. Lan, Efficient image encryption with block shuffling and chaotic map, Multimedia Tools and Applications 74 (15) (2015) 5429–5448, ISSN 15737721.

[26] A. Kanso, M. Ghebleh, A novel image encryption algorithm based on a 3D chaotic map, Communications in Nonlinear Science and Numerical Simulation 17 (7) (2012) 2943–2959, ISSN 10075704.

[27] S. Liu, J. Sun, Z. Xu, An improved image encryption algorithm based on chaotic system, Journal of Computers 4 (11) (2009) 1091–1100, ISSN 1796203X.

[28] B. Wang, Y. Xie, C. Zhou, S. Zhou, X. Zheng, Evaluating the permutation and diffusion operations used in image encryption based on chaotic maps, Optik 127 (7) (2016) 3541–3545, ISSN 00304026.

[29] G. Srividya, P. Nandakumar, A Triple-Key chaotic image encryption method, in: ICCSP 2011 - 2011 International Conference on Communications and Signal Processing, ISBN 9781424497980, 266–270, 2011.

[30] B. Norouzi, S. Mirzakuchaki, A fast color image encryption algorithm based on hyper-chaotic systems, Nonlinear Dynamics 78 (2) (2014) 995–1015, ISSN 0924090X.

[31] F. Y. Sun, S. T. Liu, Z. W. Lü, Image encryption using high-dimension chaotic system, Chinese Physics 16 (12) (2007) 3616–3623, ISSN 10091963.

[32] P. Khade, M. Narnaware, 3D Chaotic Functions for Image Encryption., IJCSI International Journal of Computer Science Issues 9 (3) (2012) 323–328, URL http://search.ebscohost.com/login.aspx?direct=true{&}profile=ehost{&}scope=site{&}authtype=crawler{&}jrnl=16940784{&}AN=77650949{&}h=Of+MgQzcarfnQimF7O12JxV4rOSvB9AgMdcJZwr8yO+aH6gJRzV3ai1mrqWGk9Ps1L4euwX88h2SaWGwWGS03Q=={&}crl=c.

[33] Y. Zhou, L. Bao, C. L. Chen, Image encryption using a new parametric switching chaotic system, Signal Processing 93 (11) (2013) 3039–3052, ISSN 01651684.

[34] A. Pisarchik, M. Zanin, Image encryption with chaotically coupled chaotic maps, Physica D: Nonlinear Phenomena 237 (20) (2008) 2638–2648, ISSN 01672789, URL http://linkinghub.elsevier.com/retrieve/pii/S0167278908001280.

[35] H. Al Haj Hassan, A. Kassem, Y. Harkouss, R. Assaf, S. El Assad, New chaotic image encryption technique, in: 2012 Symposium on Broadband Networks and Fast Internet, RELABIRA 2012, ISBN 9781467321518, ISSN 2330-4855, 103–108, 2012.

[36] Z. Parvin, H. Seyedarabi, M. Shamsi, A new secure and sensitive image encryption scheme based on new substitution with chaotic function, Multimedia Tools and Applications 75 (17) (2016) 10631–10648, ISSN 15737721.

[37] G. Ye, X. Huang, A novel block chaotic encryption scheme for remote sensing image, Multimedia Tools and Applications 75 (18) (2016) 11433–11446, ISSN 15737721.

[38] X. Wang, J. Zhao, H. Liu, A new image encryption algorithm based on chaos, Optics Communications 285 (5) (2012) 562–566, ISSN 00304018.

[39] X. Zhang, Z. Zhao, J. Wang, Chaotic image encryption based on circular substitution box and key stream buffer, Signal Processing: Image Communication 29 (8) (2014) 902–913, ISSN 09235965.

[40] K.-W. Wong, B. S.-H. Kwok, W.-S. Law, A fast image encryption scheme based on chaotic standard map, Physics Letters A 372 (15) (2008) 2645–2652, ISSN 03759601.

[41] H. Yang, K.-W. Wong, X. Liao, W. Zhang, P. Wei, A fast image encryption and authentication scheme based on chaotic maps, Communications in Nonlinear Science and Numerical Simulation 15 (11) (2010) 3507–3517, ISSN 10075704, URL http://linkinghub.elsevier.com/retrieve/pii/S1007570410000183.

[42] C.-Y. Song, Y.-L. Qiao, X.-Z. Zhang, An image encryption scheme based on new spatiotemporal chaos, Optik - International Journal for Light and Electron Optics 124 (18) (2013) 3329–3334, ISSN 00304026, URL http://linkinghub.elsevier.com/retrieve/pii/S0030402612008509.

[43] W. Zhang, H. Yu, Y. L. Zhao, Z. L. Zhu, Image encryption based on three-dimensional bit matrix permutation, Signal Processing 118 (2016) 36–50, ISSN 01651684.

[44] C. Pak, L. Huang, A new color image encryption using combination of the 1D chaotic map, Signal Processing 138 (2017) 129–137, ISSN 01651684.

[45] L. Xu, Z. Li, J. Li, W. Hua, A novel bit-level image encryption algorithm based on chaotic maps, Optics and Lasers in Engineering 78 (2016) 17–25, ISSN 01438166.

[46] S. El Assad, M. Farajallah, A new chaos-based image encryption system, Signal Processing: Image Communication 41 (2016) 144–157, ISSN 09235965.

[47] L. Bao, Y. Zhou, C. L. P. Chen, H. Liu, A new chaotic system for image encryption, in: Proceedings 2012 International Conference on System Science and Engineering, ICSSE 2012, IEEE, ISBN 9781467309455, 69–73, URL http://ieeexplore.ieee.org/document/6257151/, 2012.

[48] T. Zhang, Y. Zhou, C. L. P. Chen, A new combined chaotic system for image encryption, in: CSAE 2012 - Proceedings, 2012 IEEE International Conference on Computer Science and Automation Engineering, vol. 2, ISBN 9781467300865, 331–335, 2012.

[49] M. Kalpana, K. Ratnavelu, P. Balasubramaniam, M. Kamali, Synchronization of chaotic-type delayed neural networks and its application, Nonlinear Dynamics (2018) 1–13.

[50] G. Jakimoski, L. Kocarev, Chaos and cryptography: Block encryption ciphers based on chaotic maps, IEEE Transactions on Circuits and

Systems I: Fundamental Theory and Applications 48 (2) (2001) 163–169, ISSN 10577122.
[51] Y. Zhou, W. Cao, C. Philip Chen, Image encryption using binary bitplane, Signal Processing 100 (2014) 197–207, ISSN 01651684.
[52] K. Ratnavelu, M. Kalpana, P. Balasubramaniam, K. Wong, P. Raveendran, Image encryption method based on chaotic fuzzy cellular neural networks, Signal Processing 140 (2017) 87–96.
[53] X. Chai, X. Zheng, Z. Gan, D. Han, Y. Chen, An image encryption algorithm based on chaotic system and compressive sensing, Signal Processing 148 (2018) 124–144.
[54] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, A statistical test suite for random and pseudorandom number generators for cryptographic applications, Tech. Rep., Booz-Allen and Hamilton Inc Mclean Va, 2001.
[55] S. H. Strogatz, Nonlinear Dynamics and Chaos, vol. 48, Addison-Wesley Pub, ISBN 0738204536, 1994.
[56] B. R. Hunt, E. Ott, Defining chaos, Chaos 25 (9), ISSN 10541500.
[57] L. Kocarev, S. Lian, Chaos-based cryptography: Theory, algorithms and applications, vol. 354, Springer, 2011.
[58] M. Asgari Chenaghlu, S. Jamali, N. Nikzad Khasmakhi, A novel keyed parallel hashing scheme based on a new chaotic system, Chaos, Solitons and Fractals 87 (2016) 216–225, ISSN 09600779.
[59] S. Lynch, Dynamical systems with applications using MATLAB, Springer, 2004.
[60] M. T. Rosenstein, J. J. Collins, C. J. De Luca, A practical method for calculating largest Lyapunov exponents from small data sets, Physica D: Nonlinear Phenomena 65 (1-2) (1993) 117–134.
[61] Z. Hua, Y. Zhou, Dynamic Parameter-Control Chaotic System., IEEE Trans. Cybernetics 46 (12) (2016) 3330–3341.
[62] Y. Wu, J. P. Noonan, S. Agaian, Shannon entropy based randomness measurement and test for image encryption, arXiv preprint arXiv:1103.5520 .
[63] A. G. Weber, The USC-SIPI image database version 5, USC-SIPI Report 315 (1997) 1–24.
[64] S. Amina, F. K. Mohamed, An efficient and secure chaotic cipher algorithm for image content preservation, Communications in Nonlinear Science and Numerical Simulation 60 (2018) 12–32.
[65] E. Yavuz, M. C. Kasapbaşı, E. Yamaç, et al., A chaos-based image encryption algorithm with simple logical functions, Computers & Electrical Engineering 54 (2016) 471–483.
[66] M. Farajallah, S. El Assad, O. Deforges, Fast and secure chaos-based cryptosystem for images, International Journal of Bifurcation and Chaos 26 (02) (2016) 1650021.
[67] M. Farajallah, Chaos-based crypto and joint crypto-compression systems for images and videos, Ph.D. thesis, Universite de Nantes, 2015.
[68] Z. liang Zhu, W. Zhang, K. wo Wong, H. Yu, A chaos-based symmetric image encryption scheme using a bit-level permutation, Information Sciences 181 (6) (2011) 1171 – 1186, ISSN 0020-0255, URL http://www.sciencedirect.com/science/article/pii/S0020025510005542.
[69] L. Y. Zhang, X. Hu, Y. Liu, K.-W. Wong, J. Gan, A chaotic image encryption scheme owning temp-value feedback, Communications in Nonlinear Science and Numerical Simulation 19 (10) (2014) 3653 – 3659, ISSN 1007-5704, URL http://www.sciencedirect.com/science/article/pii/S1007570414001415.
[70] B. Schneier, The Blowfish encryption algorithm, Dr Dobb's Journal-Software Tools for the Professional Programmer 19 (4) (1994) 38–43.
[71] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson, Twofish: A 128-bit block cipher, NIST AES Proposal 15.
[72] G. Chen, Y. Mao, C. K. Chui, A symmetric image encryption scheme based on 3D chaotic cat maps, Chaos, Solitons & Fractals 21 (3) (2004) 749–761.
[73] Y. Wu, Y. Zhou, J. P. Noonan, K. Panetta, S. Agaian, Image encryption using the sudoku matrix, in: Mobile Multimedia/Image Processing, Security, and Applications 2010, vol. 7708, International Society for Optics and Photonics, 77080P, 2010.
[74] Y. Wu, J. P. Noonan, G. Yang, H. Jin, Image encryption using the two-dimensional logistic chaotic map, Journal of Electronic Imaging 21 (1) (2012) 013014.
[75] F. K. Mohamed, A parallel block-based encryption schema for digital images using reversible cellular automata, Engineering Science and Technology, an International Journal 17 (2) (2014) 85–94.
[76] X. Liao, S. Lai, Q. Zhou, A novel image encryption algorithm based on self-adaptive wave transmission, Signal Processing 90 (9) (2010) 2714–2722.
[77] Y. Wang, K.-W. Wong, X. Liao, G. Chen, A new chaos-based fast image encryption algorithm, Applied soft computing 11 (1) (2011) 514–522.
[78] W. Zhang, K.-w. Wong, H. Yu, Z.-l. Zhu, An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion, Communications in Nonlinear Science and Numerical Simulation 18 (8) (2013) 2066–2080.
[79] A. Akhshani, A. Akhavan, S.-C. Lim, Z. Hassan, An image encryption scheme based on quantum logistic map, Communications in Nonlinear Science and Numerical Simulation 17 (12) (2012) 4653–4661.
[80] K.-W. Wong, B. S.-H. Kwok, W.-S. Law, A fast image encryption scheme based on chaotic standard map, Physics Letters A 372 (15) (2008) 2645–2652.