symmetry

MDPI

Article

Chaotic Dynamical State Variables Selection Procedure Based Image Encryption Scheme

Zia Bashir [1], Jarosław Wątróbski [2,*], Tabasam Rashid [3], Sohail Zafar [3] and Wojciech Sałabun [4]

[1] Department of Mathematics, Quaid-i-Azam University, Islamabad-45320, Pakistan; ziabashir@gmail.com
[2] Faculty of Economics and Management, University of Szczecin, Mickiewicza 64, 71-101 Szczecin, Poland
[3] Department of Mathematics, University of Management and Technology, Lahore-54770, Pakistan; tabasam.rashid@umt.edu.pk (T.R.); sohailahmad04@gmail.com (S.Z.)
[4] Department of Artificial Intelligence method and Applied Mathematics in the Faculty of Computer Science and Information Technology, West Pomeranian University of Technology in Szczecin, Żołnierska 49, 71-210 Szczecin, Poland; wsalabun@wi.zut.edu.pl
* Correspondence: jaroslaw.watrobski@wneiz.pl

Received: 30 September 2017; Accepted: 6 December 2017; Published: 11 December 2017

Abstract: Nowadays, in the modern digital era, the use of computer technologies such as smartphones, tablets and the Internet, as well as the enormous quantity of confidential information being converted into digital form have resulted in raised security issues. This, in turn, has led to rapid developments in cryptography, due to the imminent need for system security. Low-dimensional chaotic systems have low complexity and key space, yet they achieve high encryption speed. An image encryption scheme is proposed that, without compromising the security, uses reasonable resources. We introduced a chaotic dynamic state variables selection procedure (CDSVSP) to use all state variables of a hyper-chaotic four-dimensional dynamical system. As a result, less iterations of the dynamical system are required, and resources are saved, thus making the algorithm fast and suitable for practical use. The simulation results of security and other miscellaneous tests demonstrate that the suggested algorithm excels at robustness, security and high speed encryption.

Keywords: chaotic maps; chaotic dynamical systems; time-varying delays

1. Introduction

Along with the swift advancement of communications and computer literacy, the use of multimedia applications has progressed quickly in all sections of society, because of their simple perspective and appealing appearance. In the meantime, the difficulties in maintaining the security of these applications have motivated a great deal of interest. Multimedia data have some inherent properties, for instance high redundancy [1] and large data capacity; therefore, the long-established encryption algorithms such as RSA, DES and AES are no longer suitable for such data. Many different image encryption algorithms, for example hash [2,3], Fibonacci [4], chaos [5–12], transform domain [13,14] and DNA [15], have been proposed to meet the security requirements.

In the theory of cryptanalysis, the security of an encryption algorithm is determined by its key space. The key space of an encryption method should be large enough to withstand a brute force attack at an obtainable computing capability. The classical permutation and diffusion encryption schemes required a higher amount of chaotic data. Generally, this need is fulfilled by one-dimensional or higher-dimensional chaotic maps, like the Arnold cat map, tent map, standard map, Beker's map, etc. Meanwhile, many encryption systems [16–23] have been successfully broken. The main reasons for the insecurity include: insufficient key space against brute force attacks, more dependence on secret keys, vulnerability against differential attack, poor sensitivity and also, with the advances in chaotic signal estimation technologies, the possibility to find chaotic orbits corresponding to the initial values

(secret keys). The usage of two independent keys in the permutation and diffusion stage, which is breakable by a known plaintext attack, can be pointed out as another drawback.

These deficiencies can be counteracted by using higher-dimensional chaotic maps or chaotic dynamical systems, as well as by some improvements, such as time-varying delays [24], couple map lattices, S-box [25], etc. In [26], Chen proposed an encryption scheme based on a dynamical state variable selection mechanism (DSVSM), which is not only fast, but also secure. Its main features are: the use of the same secret key in both the permutation and diffusion stages, involving pixels of the plain image, the use of a higher-dimensional chaotic dynamical system with great sensitivity to initial values and different key streams for different plain images by use of the same secret keys. As a result, higher security is achieved without exhausting much of the resources.

In this paper, we propose the chaotic dynamical state variable selection procedure (CDSVSP) by using a chaotic tent map. The architecture of our image encryption scheme is based on standard permutation and the diffusion system [1]. In the permutation stage, we also create the confusion with the use of the chaotic tent map, and in the diffusion stage, we use the time-varying delays. The simulation demonstrated an achievement of better results than Chen [26]. This cryptosystem can provide the security necessities recommended in [27,28] and deal with the defects present in the broken cryptosystems by making improvements in the following features:

- Chaotic state variables are generated from four-dimensional chaotic systems; a minor alteration in the secret key will not only influence the diffusion stage, but also manipulate the permutation at the same time.
- In CDSVSP, the pixels of plain images are used to choose the state variable for encryption. Thus, different key streams will be generated for each individual plain image, even if the same secret keys are used. Therefore, by encrypting individual images, the attacker is unable to extract helpful information. This characteristic guarantees the security against the known-plaintext attacks.
- In the permutation stage, the added confusion procedure can also, to some extent, create a diffusion effect. As a result, the whole effect of diffusion is increased.

The remaining part of this paper is composed as follows. The chaotic dynamic state variables selection procedure (CDSVSP) is presented in Section 2. In Section 3, the image encryption/decryption scheme is formulated and explained. Section 4 is dedicated to the numerical results and analysis of our proposed image encryption scheme. Finally, we provide the concluding remarks in Section 5.

2. Selection Procedure

In the field of cryptography, chaos is widely used. The algorithms based on chaos have exposed some exceptional features, such as complexity and security. Normally, area-preserving maps like the logistic map, bakermap and Lorenz map are used to permute the pixels of the plain image without changing the pixel values. However, in our scheme, in addition to permutation, pixel values are also changed. For this purpose, we use the chaotic tent map. The discretized tent map [29] can be defined as:

$$f(a,\rho,x) = \begin{cases} \lceil \frac{\rho}{a} x \rceil, & \text{if } 0 \le x \le a; \\ \lfloor \frac{\rho(\rho-x)}{\rho-a} x \rfloor + 1, & \text{if } a < x \le \rho, \end{cases} \tag{1}$$

where $a \in (0,\rho)$ is an integer.

In this paper, we used Lü's hyperchaotic system [30] as an example illustrating CDSVSP, as described by the following set of equations.

$$\begin{cases} \frac{dx}{dt} = 15(y-x) \\ \frac{dy}{dt} = -xz + 10y + w \\ \frac{dz}{dt} = xy - 5z \\ \frac{dw}{dt} = z - w \end{cases} \tag{2}$$

The initial system variables x_0, y_0, z_0 and w_0 act as the secret keys. In each iteration of the Lü's hyperchaotic system, we get four state variables, denoted as X, Y, Z and W, respectively. For the currently processed image with MN pixels, the pixels are arranged in a one-dimensional array $P = \{P(0), P(1), \ldots, P(MN-1)\}$ from the upper-left corner to the lower-right corner of the image. In CDSVSP, the chaotic state variables are selected with the use of the previously processed pixel and the chaotic tent map. Additionally, the value used for the first pixel is included in the secret keys. In order to demonstrate the CDSVSP properly, the following definitions are needed.

1. Let $S = \{X_i, Y_j, Z_k, W_l\}$ where X_i, Y_j, Z_k and W_l are the states of X, Y, Z and W in the i-th, j-th, k-th and l-th iteration, respectively. It should be noted that i, j, k and l do not need to be equal to each other.
2. We define $slt(L)$ as the selected variable in $\{X_i, Y_j, Z_k, W_l\}$ that will be used to generate the key stream element for $P(L)$. The decision will be made by an indicator $index(L)$, defined below:

$$index(L) = f(a, \rho, P(L-1))\%4$$

where $f(a, \rho, x)$ is a tent map and both a and ρ are parts of the secret keys.

$$slt(L) = \begin{cases} X_i & \text{if } index(L) = 0, \\ Y_j & \text{if } index(L) = 1, \\ Z_k & \text{if } index(L) = 2, \\ W_l & \text{if } index(L) = 3. \end{cases}$$

For the first pixel value, $P(-1)$ has to be set as a seed.

The procedure of CDSVSP is described as follows:

Choose i_0, j_0, k_0, l_0 sufficiently large and different from each other to act as secret keys. Iterate over Lü's hyperchaotic system $\max\{i_0, j_0, k_0, l_0\}$ times to get $S = \{X_{i_0}, Y_{j_0}, Z_{k_0}, W_{l_0}\}$ as shown in Figure 1, for the first pixel $P(0)$, and then, select the state variable for $P(0)$ by computing $index(0)$.

Based on this index, the system state is updated as follows:

$$S = \begin{cases} \{X_{i_0+1}, Y_{j_0}, Z_{k_0}, W_{l_0}\} & \text{if } index(0) = 0, \\ \{X_{i_0}, Y_{j_0+1}, Z_{k_0}, W_{l_0}\} & \text{if } index(0) = 1, \\ \{X_{i_0}, Y_{j_0}, Z_{k_0+1}, W_{l_0}\} & \text{if } index(0) = 2, \\ \{X_{i_0}, Y_{j_0}, Z_{k_0}, W_{l_0+1}\} & \text{if } index(0) = 3. \end{cases}$$

Subsequently, select the state variable from updated S for $P(1)$ by computing $index(1)$. Inductively, get the updated state variable set S for $P(n)$, and select the state variable from S by computing $index(n)$. Let us have the state $\{X_i, Y_j, Z_k, W_l\}$ for the $P(n)$, and we can assume, without loss of generality, that $index(n) = 0$. In this case, the state value X_i is chosen for $P(n)$, and the combination of state variables is reorganized to $\{X_{i+1}, Y_j, Z_k, W_l\}$. Similarly, calculate $index(n+1)$, and let $index(n+1) = 1$. The state value Y_j will be given to $P(n+1)$, and the combination state variables transform to $\{X_{i+1}, Y_{j+1}, Z_k, W_l\}$. Without loss of generality, it can be assumed that $index(n+2) = 2$. The state value Z_k will be selected for ciphering $P(n+2)$. Then, the state variable combination changes to $\{X_{i+1}, Y_{j+1}, Z_{k+1}, W_l\}$. Let us assume that $index(n+3) = 3$, without loss of generality. The state value W_l will be chosen for ciphering $P(n+3)$. Since W_l is the last element of the chaotic state W, Lü's system should be iterated enough times to produce a sufficient number of state variables for all the pixels.

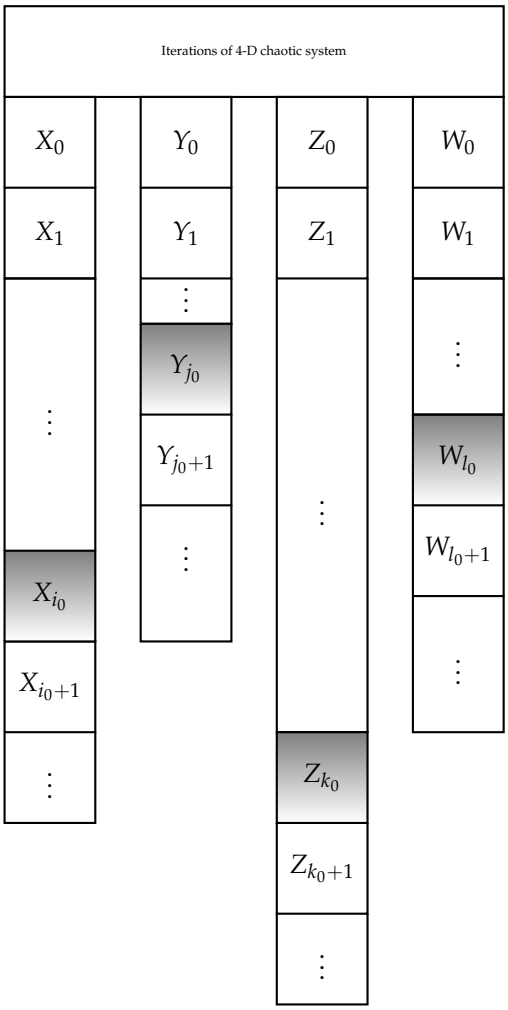

Figure 1. The starting dynamical system variables.

3. Proposed Image Encryption Scheme

The proposed algorithm is a new chaotic image encryption scheme. The entire procedure of encryption is presented concisely in the flowchart in Figure 2. Confusion and diffusion terms are required to be more specifically defined. The confusion and diffusion are properties of the operation of a secure cipher. The confusion term refers to making the relationship between the key and the ciphertext as complex and as involved as possible. Diffusion means that if we change a single bit of the image, then statistically, half of the bits in the ciphertext should change [31]. The features of the encryption procedure are as follows.

3.1. Confusion Algorithm

The key stream $k_c(n)$ is produced by using the following formula:

$$k_c(n) = 1 + mod[(abs(slt(n)) - floor(abs(slt(n)))) \times 10^{15}, 255] \quad (3)$$

where $floor(x)$ returns the nearest integer value less than or equal to x, $abs(x)$ represents the absolute value of x and $mod(x, y)$ is the remainder when x is divided by y.

First, we use a discretized tent map (1) to change the value of each pixel of the plain image. Since the grey components in 8-bit images range from 0–255, set $\rho = 255$. The discretized tent map as $f(k_c(n), 255, P(n))$ is used h_1 times to iteratively change each pixel value of the plain image.

After the above transformation, the plaintext P is converted into $P' = \{P'(0), P'(1), \ldots, P'(MN-1)\}$. Secondly, in the permutation stage, the discretized tent map $f(w, MN-1, x)$ is used h_2 times to rearrange the position of each pixel, where $w = \sum_{i=1}^{MN-1} P'(i) \bmod (MN-1)$, and the input variable $x = \{0, 1, 2, \ldots, MN-1\}$ indicates the index number of each pixel of P'. Since the tent map is a one-to-one mapping, it will give us a permutation $\sigma(x)$. Apply this permutation to P' in a way that each pixel will be sent to the place of its index value in $\sigma(x)$.

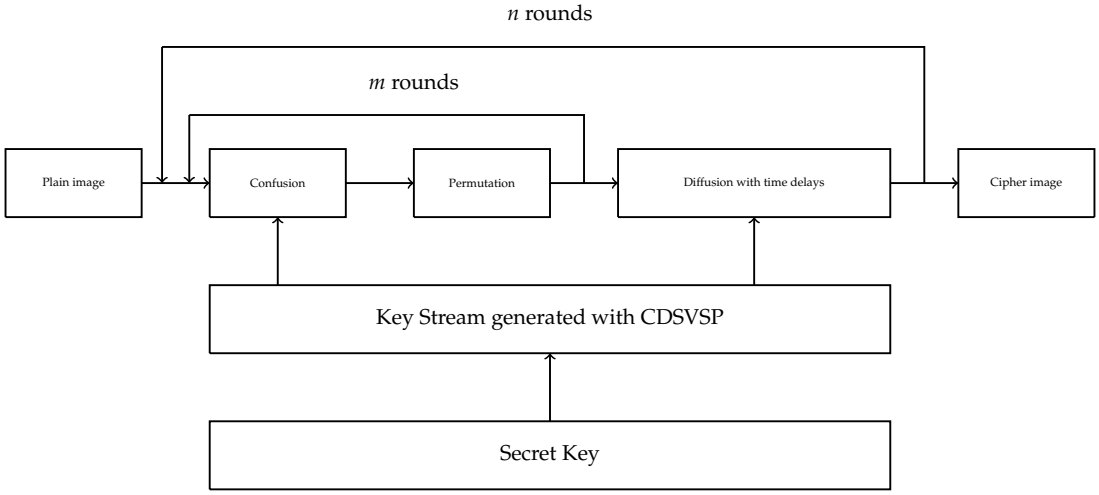

Figure 2. Flowchart of proposed scheme. CDSVSP, chaotic dynamic state variables selection procedure.

In order to test the confusion and permutation effects of our proposed image encryption scheme, we performed simulations on the Lena gray scale standard test image and its modified version, achieved by changing only one pixel value. The number of pixel changing rate (NPCR) and unified averaged changed intensity (UACI) criteria are generally useful to study the performance of our approach. The formulae to calculate $NPCR$ and $UACI$are as follows:

$$\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%$$

$$\text{UACI} = \frac{1}{L}\left[\sum_{i,j} \frac{|C(i,j) - C'(i,j)|}{255}\right] \times 100\%$$

where W represents the width and H represents the height of the image. C and C' are respectively the ciphered images before and after one pixel of the plain image is changed. $D(i,j)$ can be defined as:

$$D(i,j) = \begin{cases} 1, & \text{if } C(i,j) \neq C'(i,j); \\ 0, & \text{if } C(i,j) = C'(i,j). \end{cases}$$

The control parameters ($x_0 = -25.0$, $y_0 = 15.0$, $z_0 = -121.0$ and $w_0 = -18.0$) are used in the Lü chaotic system to generate chaotic state variables. The results, according to our confusion algorithms NPCR and UACI, are shown in Table 1 with comparison to the existing techniques.

The inspection of Table 1 will reveal the clear superiority of the proposed confusion algorithm: not only are the NPCR and UACI values much higher than Chen's result [26] and others, they are also more secure. In the sense that the attacker can by pass the permutation stage by taking the same value for all the pixels and try to break the diffusion algorithm, in the proposed algorithm, however, the pixel values are also changed, as well as permuted, so it is more secure against the known plaintext and chosen plaintext attacks.

Table 1. Test results of our proposed confusion algorithm. NPCR, number of pixel changing rate; UACI, unified averaged changed intensity.

Permutation Approaches	Rounds	NPCR	UACI
Proposed	1	99.6094	32.8120
Chen's Results [26]	1	73.38	15.87
Arnold cat map	3	3.8147×10^{-6}	1.4960×10^{-8}
Baker map	3	3.8147×10^{-6}	1.4960×10^{-8}
Standard map	3	3.8147×10^{-6}	1.4960×10^{-8}

3.2. Diffusion Algorithm

In the diffusion algorithm, the key stream $k_c(n)$ is generated by the same formula (3) as in the confusion algorithm, but the CDSVSP will be applied on the confused image.

To calculate the cipher-pixel value, we use the values of current and previous pixels according to:

$$C(n) = C(n - t(n)) \oplus k_c(n) \oplus \{(P(n) + k_c(n)) \bmod 256\} \tag{4}$$

where $P(n)$, $C(n)$ and $t(n)$ are, respectively, the currently operated pixel, output pixel and time-varying delay determined by the discretized tent map (1) as follows:

$$t(n) = f(k_c(n), \rho_t, t(n-1)).$$

Here, the initial values $t(0)$ and ρ_t are the keys. Furthermore, whenever $n - t(n) < 0$, we use a constant time delay that is a part of the secret keys.

3.3. Proposed Algorithm for Image Encryption and Decryption

The flowchart of the proposed cryptosystem is in Figure 2, and the encryption scheme is given below:

Step 1: Iterate the Lü chaotic system (2) with (x_0, y_0, z_0, w_0) for N_0 times continuously to avoid the harmful effect of the transitional procedure.
Step 2: Obtain the current state variable by means of CDSVSP. An initial value is set as the secret key for the first pixel; iterate the Lü system (2) if needed.
Step 3: Calculate the key stream for the current pixel with Equation (3).
Step 4: The discretized tent map (1) is used to change the current pixel's value.
Step 5: Go back to Step 2 until the values of all pixels are changed.
Step 6: Permute the pixels by using the discretized tent map (1) as described in the confusion algorithm.
Step 7: Repeat Steps 1–6 m times.
Step 8: Obtain the current state variable by means of CDSVSP applied on the currently processed pixel of the confused image. The initial value is set as the secret key for the first pixel.
Step 9: Calculate the key stream for the current pixel with Equation (3).
Step 10: Calculate the time-varying delays using the discretized tent map (1).
Step 11: Mask the values of the currently processed pixel using Equation (4).
Step 12: Go back to Step 8 until all pixels are encrypted.
Step 13: Repeat all these steps n times to ensure the security requirements are met.

Our proposed encryption algorithm consists of two parts: confusion and diffusion. The decryption is performed in the reverse order to the encryption. The inverse formula of masking (Equation (4)) is given in Equation (5). The discretized chaotic tent map is invertible, with the inverse given in Equation (6); for more detail of the inverse, see [29]:

$$P(n) = \{k_c(n) \oplus C(n) \oplus C(n - t(n)) + 256 - k_c(n)\} \bmod 256. \tag{5}$$

$$f^{-1}(a,\rho,x)=\begin{cases}\lfloor \frac{ax}{\rho} \rfloor, & \text{if } \lfloor \frac{ax}{\rho} \rfloor - \lceil \frac{ax}{\rho} \rceil + 1 = 0, \frac{\lfloor ax/\rho \rfloor}{a} > \frac{\lceil (a/\rho-1)x \rceil}{\rho-a};\\ \lceil (a/\rho-1)x+\rho \rceil, & \text{if } \lfloor \frac{ax}{\rho} \rfloor - \lceil \frac{ax}{\rho} \rceil + 1 = 0, \frac{\lfloor ax/\rho \rfloor}{a} \leq \frac{\lceil (a/\rho-1)x \rceil}{\rho-a};\\ \lfloor \frac{ax}{\rho} \rfloor, & \text{if } \lfloor \frac{ax}{\rho} \rfloor - \lceil \frac{ax}{\rho} \rceil + 1 = 0.\end{cases} \quad (6)$$

The decryption is done as follows.

Step 1: Iterate over the Lü chaotic system (2) with (x_0, y_0, z_0, w_0) for N_0 times continuously to avoid the harmful effect of the transitional procedure.
Step 2: Obtain the current state variable by means of CDSVSP. The initial value is known for the first pixel; iterate over the Lü chaotic system (2) if needed.
Step 3: Calculate the key stream for current pixel by Equation (3).
Step 4: Calculate the time-varying delays using the discretized tent map (1).
Step 5: Unmask the values of the currently processed pixel by using Equation (5).
Step 6: Go back to Step 2 until all pixels are undiffused.
Step 7: Apply the reverse of permutation.
Step 8: Obtain the current state variable by means of CDSVSP applied on the currently processed pixel of the image found after Step 7. The initial value is known for the first pixel.
Step 9: Calculate the key stream for the current pixel by Equation (3).
Step 10: Apply the inverse of the discretized tent map (6) to get the pixel value of the plain image.
Step 11: Go back to Step 8 until all the pixels are unconfused.
Step 12: Repeat Steps 7–11 m times.
Step 13: Repeat all these steps n times to get the plain image.

4. Analysis and Simulation Results

Numerous different experiments were performed with many standard gray scale 512 × 512-sized plain images and many encryption rounds to display the success and competence of the suggested encryption scheme. The proposed algorithm was tested in the MATLAB 2015 version with 64-bit double-precision according to IEEE [32] standard 754. The steps taken were small enough in solving the hyper chaotic Lü's dynamical system to avoid unwanted behavior [33] and degradation effects [34]. The parameters of the Lü's hyperchaotic system are $x_0 = -25.0$, $y_0 = 15.0$, $z_0 = -121.0$ and $w_0 = -18$.

4.1. Effectiveness Analysis

We took five standard gray scale test images and made their modified versions by changing the last bit of the lower right corner pixel of these test images. The proposed image cryptosystem is applied to these images. The NPCR and UACI between the encrypted test images and the cipher images of their modified versions are shown in Table 2.

Table 2. Effectiveness test results of our proposed cryptosystem.

Test Images	1 Round		1 Round		2 Rounds	
	Permutation	**Encryption**	**Overall**	**Encryption**	**Overall**	**Encryption**
	NPCR	UACI	NPCR	UACI	NPCR	UACI
Lena	99.6094%	32.8120%	99.6136%	33.4880%	99.6002%	33.4630%
Baboon	99.6090%	32.8388%	99.6029%	33.4859%	99.6235%	33.4671%
Peppers	99.6132%	33.2538%	99.6185%	33.4838%	99.6269%	33.4050%
Bridge	99.5766%	33.5782%	99.6082%	33.5055%	99.6159%	33.5187%
Boat	99.5953%	32.6331%	99.6052%	33.5188%	99.6128%	33.4390%

Hence, a single round of encryption is enough to get effective cipher images. Furthermore NPCR > 99.6% and UACI > 33.4 in all cases, proving that the proposed image encryption scheme is secure and protected against many attacks like differential attack. A change of a single pixel will result

in obtaining entirely different cipher images, so that a known plaintext attack is successfully defended. The above simulation results proved the effectiveness of our proposed scheme.

4.2. Efficiency Comparisons

For an efficient encryption algorithm, NPCR and UACI should be greater than 99.6% and 33.4%, respectively. The efficiency of any algorithm is measured in terms of achieving these levels with minimum resources. Wong et al. [12,35] pointed out that efficiency is reflected by the average chaotic variables and average quantization operations, required in the encryption process. Thus, a comparison of the efficiency between the proposed encryption algorithm and five typical encryption algorithms [11,26,36–38] based on these parameters is presented in Table 3.

Table 3. Efficiency analysis of the image encryption schemes to achieve a satisfactory security level.

	NPCR (%)	UACI (%)	Average Encryption Rounds	Average Required Chaotic Variables	Average Required Quantization Operations
Proposed	>99.6	>33.4	1	1.002	2
Ref. [26]	>99.6	>33.4	1	1.004	2
Ref. [36]	>99.6	>33.4	1	4	2
Ref. [11]	>99.6	>33.4	3	9	3
Ref. [37]	>99.6	>33.4	2	7	2
Ref. [38]	>99.6	>33.4	2	6	2

The proposed encryption algorithm needs only one round of encryption to achieve NPCR >99.6 and UACI >33.4, so for Chen's algorithm [26] and Fu's algorithm [36]. However, it is more secure, as already discussed in Section 3.1. For encryption of the gray scale 512 × 512 sized image, only 65,650 iterations of Lü's hyperchaotic system are needed. Thus 65,650 × 4 = 262,600 state variables are used to generate the key stream, and therefore, 1.002 chaotic variables are required to cypher each pixel on average. Hence, the proposed encryption algorithm is better in comparison to [11,26,36–38], in one way or another.

4.3. Key Space Analysis

The key is a very essential aspect of every cryptosystem. An algorithm is only as secure as its key. Even if an algorithm is very strong and well designed, if the key is chosen poorly or the key space is too small, the cryptosystem will be broken eventually. The strength of any cryptographic algorithm depends on the size of its key space to make brute force attack unfeasible. In our proposed algorithm, the secret key consists of four parameters X_0, Y_0, Z_0 and W_0 of the Lü's chaotic system. For the simulation of the proposed scheme, we use 64-bit double precision. According to IEEE floating point standards, the computational accuracy is 10^{15}. As a result, the total number of likely values of the secret key is around $10^{15} \times 4$, which is large enough to resist a brute-force attack.

4.4. Key Sensitivity Analysis

The key sensitivity analysis guarantees the security of the cryptosystems against the brute-force attack. For any cryptosystem, the key sensitivity means that the two cipher images should be entirely independent of each other if the attacker uses two slightly different keys to encrypt the same plain image. To assess the key sensitivity, at first, we did the single-round encryption with keys ($x_0 = -25.0$, $y_0 = 15.0$, $z_0 = -121.0$ and $w_0 = -18$). Then, we added 10^{-14} to one of the parameters, whilst all others stayed unchanged, and we performed the encryption process again. The corresponding cipher images and the differential images are shown in Figure 3. The differences between the corresponding cipher images are computed and given in Table 4. The results clearly demonstrate that the cipher images have no relation between each other, and there is no considerable correlation that could be observed in the differential images.

Table 4. Differences between cipher images produced by slightly different keys.

Figures	Encryption Keys				Differences Ratio
	x_0	y_0	z_0	w_0	
Lena 1	$-25+10^{-14}$	15	-121	-18	99.5903%
Lena 2	-25	$15+10^{-14}$	-121	-18	99.6025%
Lena 3	-25	15	$-121+10^{-14}$	-18	99.6220%
Lena 4	-25.0	15	-121	$-18+10^{-14}$	99.6048%
Average					99.6049%

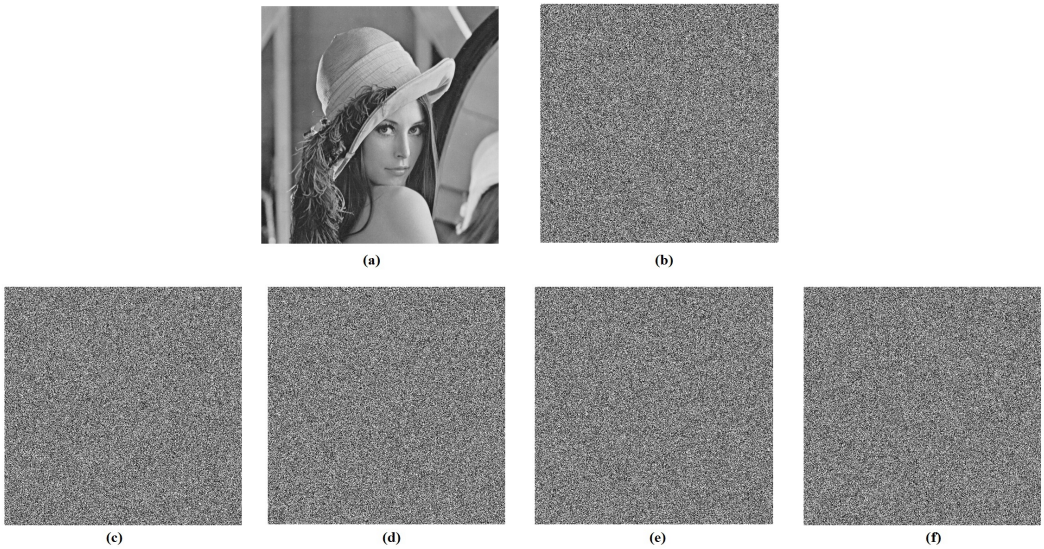

Figure 3. Key sensitivity in the first case: (**a**) plain image; (**b**) cipher image ($x_0 = -25.0$, $y_0 = 15.0$, $z_0 = -121.0$, $w_0 = -18$); (**c**) cipher image ($x_0 = -25.000000000000001$, $y_0 = 15.0$, $z_0 = -121.0$, $w_0 = -18$); (**d**) cipher image ($x_0 = -25$, $y_0 = 15.000000000000001$, $z_0 = -121$, $w_0 = -18$); (**e**) cipher image ($x_0 = -25.0$, $y_0 = 15.0$, $z_0 = -121.000000000000001$, $w_0 = -18$); (**f**) cipher image ($x_0 = -25.0$, $y_0 = 15.0$, $z_0 = -121.0$, $w_0 = -18.000000000000001$).

4.5. Histogram Analysis

An image histogram shows the pixels distribution in an image by plotting all the pixels. Here, we use the well-known 'Lena.jpg' 512 × 512 pixel plain image. The plain and cipher images are shown in Figure 4a,c, respectively. Their corresponding histogram analyses are shown in Figure 4b,d. It is easy to notice that the histogram of the cipher (encrypted) image is uniformly distributed and completely different from that of the plain (Lena gray scale) image.

4.6. Correlation Analysis

In the first step, the number of pairs of neighboring pixels should be chosen. Based on the literature study, we can assume that the four most commonly-used variants are 3000, 4000, 8000 and 10,000 randomly-selected pairs [39–41]. We have conducted research on all of these variants; nevertheless, the best results were obtained for the variant containing 3000 pairs. The comparison of the results is presented in Table 5. Due to having the best results, the 3000-pair variant will be presented in detail. If less than 3000 pairs were selected, the correlation between the neighboring pixels would not be exhibited comprehensively. A set of 3000 randomly-selected pairs of neighboring pixels (in the vertical, horizontal and diagonal directions) was collected from the plain and ciphered images, and the correlation coefficients of each two neighboring pixels were calculated according to the following formulas:

$$C_{x,y} = \frac{cov(x,y)}{\sqrt{D(x)D(y)}},$$

where:

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)),$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N}x_i,$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2.$$

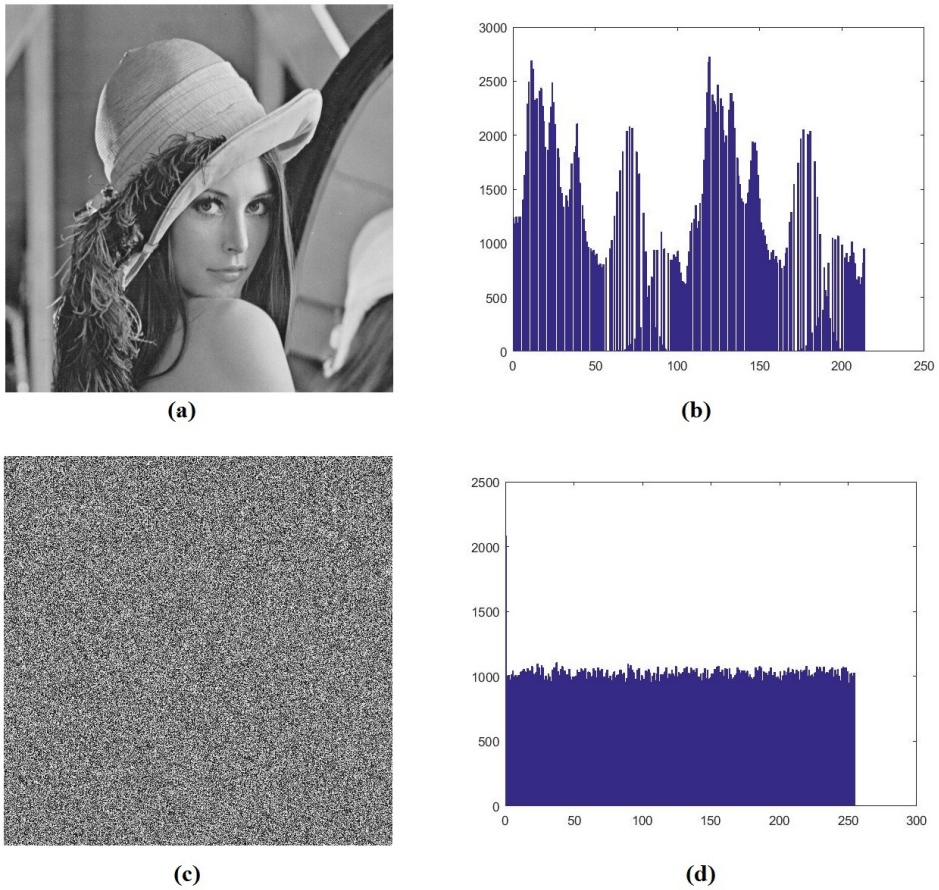

Figure 4. Histogram of images: (**a**) Lena gray scale image; (**b**) histogram of Lena image; (**c**) cipher (encrypted) image ($x_0 = -25.0$, $y_0 = 15.0$, $z_0 = -121.0$, $w_0 = -18$); (**d**) histogram of cipher (encrypted) image.

The correlation coefficients of the adjacent pixels in the plain image and its cipher image are listed in Table 5. Both the calculated correlation coefficients and Figures 5 and 6 indicate that the correlation of the two adjacent pixels of the plain image is large, while that of the encrypted image is very small, so the encryption effect is satisfactory.

Table 5. Correlation coefficients of adjacent pixels at the first iteration.

	3000 Pairs		4000 Pairs	
Direction	**Plain Image**	**Cipher Image**	**Plain Image**	**Cipher Image**
Horizontal	0.97454	−0.00932	0.919702	0.020973
Vertical	0.986736	0.010248	0.958690	−0.004789
Diagonal	0.959988	−0.005223	0.893104	0.032478
	8000 Pairs		**10,000 Pairs**	
Direction	**Plain Image**	**Cipher Image**	**Plain Image**	**Cipher Image**
Horizontal	0.9239702	0.0145748	0.919298	−0.017349
Vertical	0.9540613	−0.000374	0.954782	0.0054973
Diagonal	0.9004525	-0.000569	0.892229	0.0125824

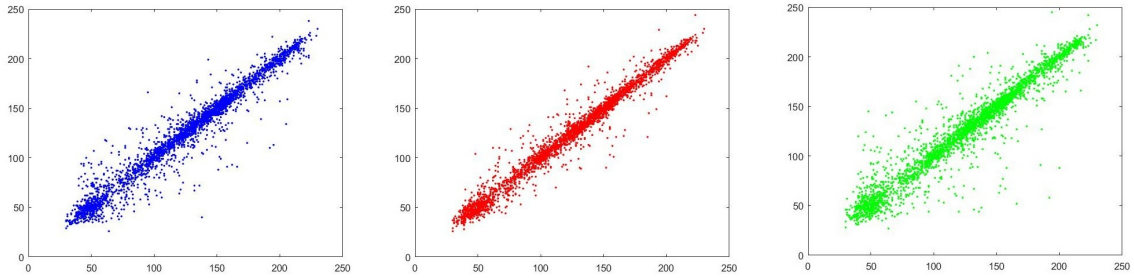

Figure 5. Correlation of 3000 adjacent random pixels of the plain image: (**a**) horizontal adjacent pixels; (**b**) vertical adjacent pixels; (**c**) diagonal adjacent pixels.

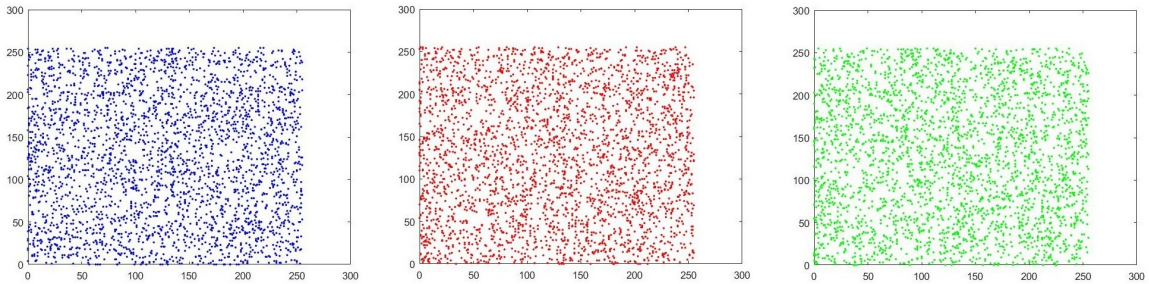

Figure 6. Correlation of 3000 adjacent random pixels of the cipher image: (**a**) horizontal adjacent pixels; (**b**) vertical adjacent pixels; (**c**) diagonal adjacent pixels.

4.7. Entropy Measure Analysis

In 1949, Shannon found the unpredictability and randomness of an information source, called information entropy [42]. It is a mathematical property. The entropy measure $H(s)$ of a message source s is defined as:

$$H(s) = -\sum_{i=0}^{2^N-1} P(s_i) \log_2 P(s_i),$$

where N is the number of bits to represent the symbol s_i and $P(s_i)$ is the probability of the symbol s_i. The entropy measure is N for a truly random source consisting of 2^N symbols. The ideal entropy for a 256 gray scale level image is eight. The lesser the entropy, the lesser the randomness and security. Information entropy was calculated for six different 256 gray scale test images of 512 × 512 in size before and after the first round of encryption. The values are given in Table 6. The analysis of the

values from Table 6 shows that the entropy values of the cipher images are very close to eight, which guarantees the randomness and unpredictability of the cipher image.

Table 6. Entropy measures of plain images and cipher images.

Test Images	Plain Image	Cipher Image
Lena	7.4455	7.9994
Baboon	7.3713	7.9992
Peppers	7.5800	7.9993
Bridge	5.7922	7.9993
Boat	7.1914	7.9992

5. Conclusions

The main contribution of the paper is to propose an image cryptosystem utilizing a four-dimensional chaotic system in order to get highly secure results. For this purpose, a new chaotic dynamic state variables selection procedure (CDSVSP) was developed. Low-dimensional chaotic systems are a useful tool for achieving low complexity and relatively small key space, yet obtaining high encryption speed at the same time.

The paper presents the theoretical foundations of the proposed approach, ensuring a very high level of security of the presented system. The chaotic sequence can be utilized to produce a key stream, which is then used in the confusion and diffusion stages. Furthermore, the discretized tent map increases the security even more by changing the pixel values, which creates some sort of diffusion. The proposed approach is also consistent with the current research trends on increasing the level of system security.

We verified the security of the image encryption scheme against numerous attacks, which allowed us to reach the conclusion that our image encryption scheme is highly secure and most suitable for image encryption. The results of the presented numerical example show that the entropy values of the cipher images ensure the randomness and unpredictability of the cipher images.

During the research, some possible areas of improvement have been identified. The potential future work directions could focus on:

- practical utilization of the proposed procedure and system;
- broader comparison of the obtained results with other approaches;
- searching for possibilities to increase the level of system security even further.

Acknowledgments: The authors are thankful to the reviewers for their comments and suggestions to improve the quality of the manuscript.

Author Contributions: This paper is a result of the common work of the authors in all aspects.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fridrich, J. Symmetric ciphers based on two-dimensional chaotic maps. *Int. J. Bifurc. Chaos* **1998**, *8*, 1259–1284.
2. Cheddad, A.; Condell, J.; Curran, K.; McKevitt, P. A hashbased image encryption algorithm. *Opt. Commun.* **2010**, *283*, 879–893.
3. Tarmissi, K.; Hamza, A.B. Information-theoretic hashing of 3D objects using spectral graph theory. *Expert Syst. Appl.* **2009**, *36*, 9409–9414.
4. Zhou, Y.; Panetta, K.; Agaian, S.; Chen, C.L.P. Image encryption using $p-$Fibonacci transform and decomposition. *Opt. Commun.* **2012**, *285*, 594–608.
5. Bashir, Z.; Rashid, T.; Zafar, S. Hyperchaotic dynamical system based image encryption scheme with time-varying delays. *Pac. Sci. Rev. A Nat. Sci. Eng.* **2016**, doi:10.1016/j.psra.2016.11.003.
6. Elsheh, E.; Hamza, A.B. Secret sharing approaches for 3D object encryption. *Expert Syst. Appl.* **2011**, *38*, 13906–13911.

7. Chen, G.; Mao, Y.; Chui, C.K. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Solitons Fractals* **2004**, *21*, 749–761.
8. Fu, C.; Lin, B.; Miao, Y.; Liu, X.; Chen, J. A novel chaos-based bit-level permutation scheme for digital image encryption. *Opt. Commun.* **2011**, *284*, 5415–5423.
9. García-Martínez, M.; Ontañón-Garcíab, L.J.; Campos-Cantóna, E.; Čelikovský, S. Hyperchaotic encryption based on multi-scroll piecewise linear systems. *Appl. Math. Comput.* **2015**, *270*, 413–424.
10. Lian, S.; Sun, J.; Wang, Z. A block cipher based on a suitable use of the chaotic standard map. *Chaos Solitons Fractals* **2005**, *26*, 117–129.
11. Mao, Y.; Chen, G.; Lian, S. A novel fast image encryption scheme based on 3D chaotic baker maps. *Int. J. Bifurc. Chaos* **2004**, *14*, 3613–3624.
12. Wong, K.; Kwok, B.; Law, W. A fast image encryption scheme based on chaotic standard map. *Phys. Lett. A* **2008**, *372*, 2645–2652.
13. Lang, J. Image encryption based on the reality-preserving multipleparameter fractional fourier transform. *Opt. Commun.* **2012**, *285*, 2584–2590.
14. Zhou, N.; Wang, Y.; Gong, L. Novel optical image encryption scheme based on fractional mellin transform. *Opt. Commun.* **2011**, *284*, 3234–3242.
15. Liu, H.; Wang, X. Image encryption using dna complementary rule and chaotic maps. *Appl. Soft Comput.* **2012**, *12*, 1457–1466.
16. Arroyo, D.; Li, S.; Amigo, J.; Alvarez, G.; Rhouma, R. Comment on image encryption with chaotically coupled chaotic maps. *Phys. D Nonlinear Phenom.* **2010**, *239*, 1002–1006.
17. Cokal, C.; Solak, E. Cryptanalysis of a chaos-based image encryption algorithm. *Phys. Lett. A* **2009**, *373*, 1357–1360.
18. Li, C.; Li, S.; Lo, K. Breaking a modified substitutiondiffusion image cipher based on chaotic standard and logistic maps. *Commun. Nonlinear Sci. Numer. Simul.* **2011**, *16*, 837–843.
19. Li, C.; Li, M.A.S.; Nunez, J.; Alvarez, G.C.G. On the security defects of an image encryption scheme. *Image Vis. Comput.* **2009**, *27*, 1371–1382.
20. Li, C.; Liu, Y.; Xie, T.; Chen, M. Breaking a novel image encryption scheme based on improved hyperchaotic sequences. *Nonlinear Dyn.* **2013**, *73*, 2083–2089.
21. Rhouma, R.; Solak, E.; Belghith, S. Cryptanalysis of a new substitutiondif-fusion based image cipher. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 1887–1892.
22. Solak, E.; Cokal, C.; Yildiz, O.; Biyikoglu, T. Cryptanalysis of Fridrich's chaotic image encryption. *Int. J. Bifurc. Chaos* **2010**, *20*, 1405–1413.
23. Wang, K.; Pei, W.; Zou, L.; Song, A.; He, Z. On the security of 3D cat map based symmetric image encryption scheme. *Phys. Lett. A* **2005**, *343*, 432–439.
24. Tang, Y.; Wang, Z.; Fang, J.A. Image encryption using chaotic coupled map lattices with time-varying delays. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 2456–2468.
25. Hussain, I.; Shah, T.; Gondal, M.A. Application of S-box and chaotic map for image encryption. *Math. Comput. Model.* **2013**, *57*, 2576–2579.
26. Chen, J.; Zhu, Z.; Fu, C.; Yu, H.; Zhang, L. A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism. *Commun. Nonlinear Sci. Numer. Simul.* **2015**, *20*, 846–860.
27. Alvarez, G.; Li, S. Some basic cryptographic requirements for chaos-based cryptosystem. *Int. J. Bifurc. Chaos* **2006**, *16*, 2129–2151.
28. Furht, B.; Kirovski, D. *Multimedia Security Handbook*; Ch. Chaos-Based Encryption for Digital Images and Videos; CRC Press: New York, NY, USA, 2004.
29. Masuda, N.; Aihara, K. Cryptosystems with discretized chaotic maps. *IEEE Trans. Circ. Syst. I* **2002**, *49*, 28–40.
30. Chen, A.; Lu, J.; Yu, S. Generating hyperchaotic Lü attractor via state feedback control. *Physica A* **2006**, *364*, 103–110.
31. Stallings, W. *Cryptography and Network Security: Principles and Practices*; Pearson Education: London, UK, 2006.
32. IEEE Computer Society. IEEE standard for binary floating-point arithmetic. *ANSI/IEEE Std.* **1985**, 754–1985, doi:10.1109/IEEESTD.1985.82928.
33. Öztürk, I.; Kiliç, R. Cycle lengths and correlation properties of finite precision chaotic maps. *Int. J. Bifurc. Chaos* **2014**, *24*, 1450107.

34. Li, S.; Chen, G.; Mou, X. On the dynamical degradation of digital piecewise linear chaotic maps. *Int. J. Bifurc. Chaos* **2005**, *15*, 3119–3151.
35. Wong, K.; Kwok, B.; Yuen, C. An efficient diffusion approach for chaos-based image encryption. *Chaos Solitons Fractals* **2009**, *41*, 2652–2663.
36. Fu, C.; Chen, J.J.; Zou, H.; Meng, W.H.; Zhan, Y. F.; Yu, Y.W. A chaos-based digital image encryption scheme with an improved diffusion strategy. *Opt. Express* **2012**, *20*, 2363–2378.
37. Wang, Y.; Wong, K.; Liao, X.; Xiang, T.; Chen, G. A chaos-based image encryption algorithm with variable control parameters. *Chaos Solitons Fractals* **2009**, *41*, 1773–1783.
38. Zhang, W.; Wong, K.; Yu, H.; Zhu, Z. An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion. *Commun. Nonlinear Sci. Numer. Simul.* **2013**, *18*, 2066–2080.
39. Belazi, A.; El-Latif, A.A.A.; Diaconu, A.V.; Rhouma, R.; Belghith, S. Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms. *Opt. Lasers Eng.* **2017**, *88*, 37–50.
40. Li, Y.; Wang, C.; Chen, H. A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. *Opt. Lasers Eng.* **2017**, *90*, 238–246.
41. Zhou, N.; Zhang, A.; Wu, J.; Pei, D.; Yang, Y. Novel hybrid image compression–encryption algorithm based on compressive sensing. *Opt. Int. J. Light Electron Opt.* **2014**, *125*, 5075–5080.
42. Shannon, C. Communication theory of secrecy systems. *Bell Syst. Tech. J.* **1949**, *18*, 656–715.