A Novel Cryptosystem Using Dynamics Perturbation of Logistic Map

Thang Manh Hoang[*]*, Hoang Xuan Thanh*

Hanoi University of Science and Technology - No. 1, Dai Co Viet, Hai Ba Trung, Hanoi, Viet Nam

Received: February 14, 2019; Accepted: November 28, 2019

Abstract

In this paper, a novel cryptosystem is proposed using the method of dynamics perturbation to the Logistic map. The perturbation is carried out by changing the value of its control parameter in bit level after every iteration during encryption and decryption. Consequently, the dynamics of Logistic map becomes non-stationary, and it helps to resist the statistical attacks. Moreover, the key space is expanded significantly. The bit distribution balancing is proposed to have better statistical properties of ciphertext. The example will show the effectiveness of the proposed cryptosystem and the simulation results are compared with those in other works published recently.

Keywords: chaotic cryptosystem, Logistic map

1. Introduction

Recently, the chaotic cryptosystem has been studied extensively [1]. The chaotic cryptography employs chaotic dynamics generated by chao systems to have the complexity for encryption; this is the alternative approach to encrypt data, beside the conventional approach of using numerical complexity [2]. The main advantages of using chaos over the conventional approach is the simpleness in design and the elastic keyspace. However, the disadvantage so far for chaotic crytosystem is the lack of provability [3]. Extensive study has been pursuited to make chaotic cyrpstosystem reliable and applicable.

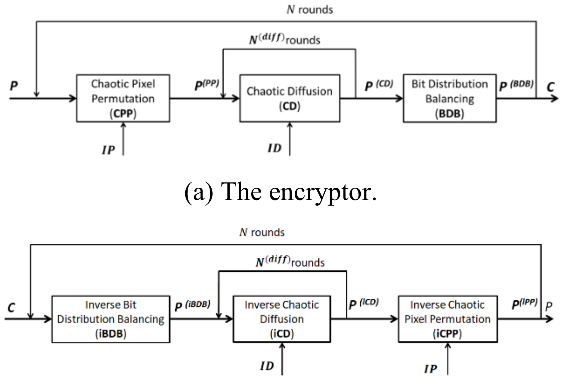
There are few main ways in employing chaotic systems for cryptography. Those are that (i) chaotic state values are used as a random sequence for encryption [4]; (ii) chaotic dynamics involves in encryption and decryption by means of perturbing/modulating on the initial vector (IV) and/or on the control parameters [5]. The perturbation is known as changing the value of state variable or of control parameter during iterations. The proposed cryptosystem in this paper is categorized into the second way.

More recently, the statistical properties of bits in bit layers of image have been analyzed [6]. It shows that the statistics of bits in different bit layers of image should be balanced, and the method of bit balancing was proposed [7]. In that work, the statistical properties of ciphertext produced by a cryptosystem with the inclusion of bit balancing is improved significantly.

In this work, a novel cryptosystem is proposed with the use of Logistic map. The structure of the cryptosystem is the Unified model [2], with permutation and diffusion. The Logistic map is employed for both permutation and diffusion. All the chaotic value and the control parameter of Logistic map are represented in the format of fixed point. The perturbation is made in bit level to the control parameter. In addition, the bit distribution balancing is proposed to improve the statistical properties in the ciphertext. The simulation result shows the effectiveness of the proposed system and the simulation results are compared with those in other works published recently.

2. The proposed cryptosystem

The structure of the proposed cryptosystem is in the form of Unified model as shown in Fig. 1. It consists of the permutation, the diffusion and the bit distribution balancing.


N rounds
$N^{(diff)}$rounds
P
Chaotic Pixel Permutation (**CPP**)
$P^{(PP)}$
Chaotic Diffusion (**CD**)
$P^{(CD)}$
Bit Distribution Balancing (**BDB**)
$P^{(BDB)}$
C
IP
ID
N rounds
$N^{(diff)}$rounds
C
Inverse Bit Distribution Balancing (**iBDB**)
$P^{(iBDB)}$
Inverse Chaotic Diffusion (**iCD**)
$P^{(iCD)}$
Inverse Chaotic Pixel Permutation (**iCPP**)
$P^{(iPP)}$
P
ID
IP

(a) The encryptor.

(b) The decryptor.

Fig. 1. Structure of cryptosystem

[*] Corresponding author: Tel.: (+84) 988.802.694
Email: thang.hoangmanh@hust.edu.vn

The encryptor and decryptor employ the Logistic map for the permutation and diffusion processes.

The Logistic map is a simplest, discrete, 1D chaotic system with the only one control parameter. The equation for the Logistic map is

$$X_{n+1} = rX_n(1 - X_n), \qquad (1)$$

where r is the control parameter. Chaotic dynamics of Logistic map exibits for the value of control parameter roughly in the range of 3.56 to 4.0. The most chaotic behavior is obtained at $r = 4.0$. The range of chaotic state variable is $x_n \in (0,1)$.

Let us consider the 8-bit grayscale image with M rows and N columns of pixels, i.e. the size of $M \times N$. The value of pixels at the position (x, y) in the image is $p(x, y)$ and it is represented by 8 bits, i.e. $p(x, y) = b_7b_6b_5b_4b_3b_2b_1b_0$. There, b_7 and b_0 are most and least significant bit, respectively. The permutation shuffles the pixels and the diffusion scrambles the value of pixels.

2.1. The encryptor

As shown in Fig. 1(a), the encryptor consists the block of chaotic pixel permutation (CPP), chaotic diffusion (CD) and bit distribution balancing (BDB). The permutation, diffusion and whole processes can be iterated more than one time.

2.1.1. Chaotic pixel permutation (CPP)

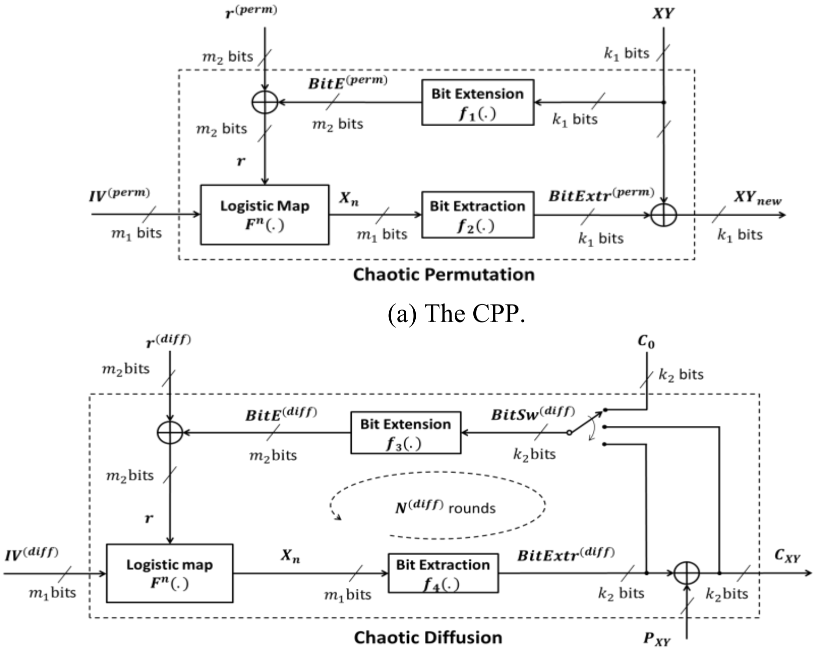
The CPP carries out the pixel permutation over the image space. The value of pixel at (*x,y*) will be exchanged with that at (x_{new}, y_{new}). The structure of CPP is illustrated at Fig. 2(a). The position information of pixels (x, y) and (x_{new}, y_{new}) are represented by the bit sequences XY and XY_{new}. The length of bit sequences is dependent on the size of image, i.e. $k_1 = \log_2 M \times N$.

In this work, the representation for these values is the form of fixed point. As shown in Fig.2(a), the number of bits representing for chaotic value X_n and control parameter r is m_1 and m_2. Due to the range of those values of the Logistic map, for the chaotic value, one bit is for the integer part and $(m_1 - 1)$ bits for the fractional part; for the value of control parameter, two bits for the integer part and $(m_2 - 2)$ bits for the fractional part.

In the permutation as shown in Fig. 2(a), the value of control parameter r of the Logistic map is m_2 bits and found by

$$r = r^{(perm)} \oplus BitE^{(perm)}, \qquad (2)$$

with $BitE^{(perm)} = f_1(XY)$. $f_1(.)$ is the extension function, in which the positions of k_1 bits of XY to be extended relatively to m_2 bits $BitE^{(perm)}$ is defined by the list of bit positions $Q_1^{(perm)}$. The value of $r^{(perm)}$ and the list $Q_1^{(perm)}$ are chosen so that the resultant value of r is within the range, e.g. 3.56 to 4.0, to guarante the exhibition of chaotic behavior. It is clear that different values of r are used for shuffling different pixels.



(a) The CPP.

(b) The CD.

Fig. 2. Structure of CPP and CD.

Here, the Logistic map can be iterated n times as $X_{n+1} = F^n(X_n)$ with the $IV^{(perm)}$is the initial value. New position of a pixel is

$$XY_{new} = XY \oplus BitExtr^{(perm)}, \quad (3)$$

where bits of $BitExtr^{(perm)}$ is extracted from from X_n, or $BitExtr^{(perm)} = f_2(X_n)$. The extraction function $f_2(.)$ specifies which positions of k_1 bits from X_n to be extracted for $BitExtr^{(perm)}$ by the list of bit positions $Q_2^{(perm)}$.

2.1.2. Chaotic diffusion (CD)

The chaotic diffusion performs on pixel data continually as shown in Fig.2(b). The control parameter of the Logistic map is represented by m_2 bits and calculated by

$$r = r^{(diff)} \oplus BitE^{(diff)}. \quad (4)$$

The Logistic map can be itereated m times as $X_{n+1} = F^m(X_n)$ with the initial value $IV^{(diff)}$. Let us define the present diffusion round $n^{(diff)}$ $(1 \le n^{(diff)} \le N^{(diff)})$. $BitE^{(diff)}$ is obtained by the bit extension function $f_1(.)$ with the input $BitSw^{(diff)}$, and

$$BitSw^{(diff)} = \begin{cases} C_0 \text{ for } n^{(diff)} = 1 \text{ and } p(x,y) \text{ with } x = 0 \text{ and } y = 0 \\ C_{XY} \text{ for } n = 1 \text{ and } p(x,y) \text{ with } x \ne 0 \text{ or } y \ne 0 \\ BitExtr^{(diff)} \text{ for } 1 < n^{(diff)} \le N^{(diff)} \text{ and } p(x,y) \text{ with } \forall x,y \end{cases}. \quad (5)$$

There, $BitExtr^{(diff)}$ is extracted from X_n, or $BitExtr^{(diff)} = f_2(X_n)$. The rules for the bit extension $f_1(.)$ and bit extraction $f_2(.)$ are defined by the lists $Q_1^{(diff)}$and $Q_2^{(diff)}$, respectively. The value of diffused pixels C_{XY} is

$$C_{XY} = P_{XY} \oplus BitExtr^{(diff)}. \quad (6)$$

$C_0$is the initial ciphertext byte.

2.1.3. Bit distribution balancing (BDB)

The bit distribution balancing is to balance the number of bits 0 and 1. This help to reduce the number of encryption rounds [7]. In this research, the proposed BDB scheme dealing with bits of pixel is introduced. The scheme is based on the property of natural images that the bit distribution at the lower layers is more bias in compared with that at the higher layers. For the 8 bits grayscale image, the bits at the higher layers is replaced by

$$b_i' = \begin{cases} b_i \text{ for } i = 0; \\ b_0 \oplus b_i \text{for } 1 \le \mathrm{i} \le 7 \end{cases} \quad (7)$$

The index i denotes for the layer of bits.

2.2. The decryptor

As shown in Fig. 1b, the decryptor consists of inverse CPP (iCPP), inverse CD (iCD), and inverse BDB (iBDB), in which these functions are arranged in the reverse way in compared with that in the encryptor. Among them, only the iCD is with structure different from CD of the encryptor, while the other functions iCPP and iBDB are exactly the same as CPP and BDB in the encryptor. This is due to the way of calculating the new pixel position and of using XOR operator.

The detailed structure of iCD is illustrated in Fig. 3. The main difference between the iCD and the CD in Fig. 2b is the block Z^{-1} to make C_{XY} delayed for calculating the future pixel. The value of $BitSw^{(diff)}$ becomes

$$BitSw^{(diff)} = \begin{cases} C_0 \text{ for } n^{(diff)} = 1 \text{ and } p(x,y) \text{ with } x = 0 \text{ and } y = 0 \\ C_{XY}^{-1} \text{ for } n^{(diff)} = 1 \text{ and } p(x,y) \text{ with } x \ne 0 \text{ or } y \ne 0 \\ BitExtr^{(diff)} \text{ for } 1 < n^{(diff)} \le N^{(diff)} \text{ and } p(x,y) \text{ with } \forall x,y \end{cases}. \quad (8)$$

The constraint for the number of bits in the cryptosystem will be considered.

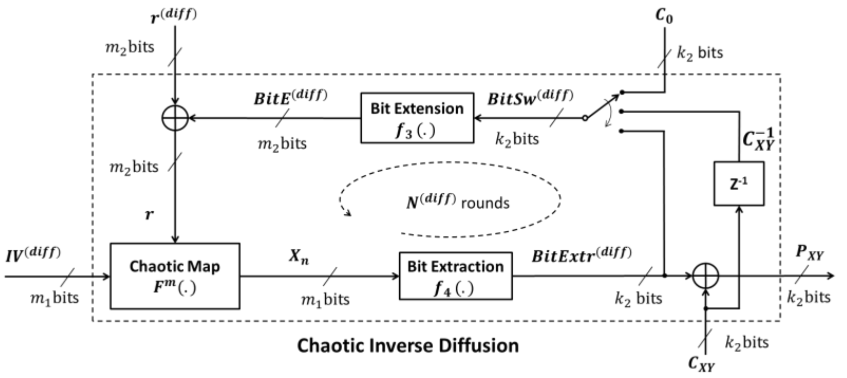


Fig. 3. Structure of iCD.

3. Simulation results

In order to illustrate the operation of the cryptosystem, the example is simulated for 8 bits gray scale images with the size of 256×256, or $k_1 = \log_2 256 \times 256 = 16$ bits and $k_2 = 8$ bits. Tables 1 and 2 show the number of bits and the chosen value for parameters, respectively.

Table 1. The number of bits representing for data.

Parameters	#bits	Fixed point format
m_1	32	1.31
m_2	33	2.31

Table 2. The value of parameters and number of bits for representation.

Parameters	Chosen value	Represented by #bits
$r^{(perm)}$	3.625	33
$r^{(diff)}$	3.625	33
$IV^{(perm)}$	0.0123456789	32
$IV^{(diff)}$	0.9876543210	32
C_0	123	8
Total no. of bits		**138**

The value of other parameters for the simulation is $N^{(diff)} = 1$, $N = 1$, $n = 1$, and $m = 1$.

In order to guarantee the chaotic behavior exhibited by the Logistic map, $r^{(perm)} = r^{(diff)} = 3.625$ is chosen, so its binary representation is $111010000000000000000000000000000$. The bit positions in $r^{(perm)}$ and $r^{(diff)}$ to be perturbed to produce r are from b_0 to b_{28}. As a result, the value range of r for the Logistic map in both the CPP and the CD is from 3.625 to 3.99999 (2^{-31} less than 4.0).

The extension functions $f_1(.)$ and $f_3(.)$are filled zeros to the bit positions rather than those indicated in $Q_1^{(perm)}$ and $Q_1^{(diff)}$. Moreover, the extraction functions $f_2(.)$ and $f_4(.)$ collect bits from X_n with the order of bits defined in $Q_2^{(perm)}$ and $Q_2^{(diff)}$.

It is noted that the order of bits specified by $Q_1^{(perm)}$, $Q_1^{(diff)}$, $Q_2^{(perm)}$, and $Q_2^{(diff)}$ will make the key space expanded significantly. In this example, the sizes are $|Q_1^{(perm)}| = 16$, $|Q_1^{(diff)}| = 8$, $|Q_2^{(perm)}| = 16$, and $|Q_2^{(diff)}| = 8$. The chosen order is displayed in Table 3.

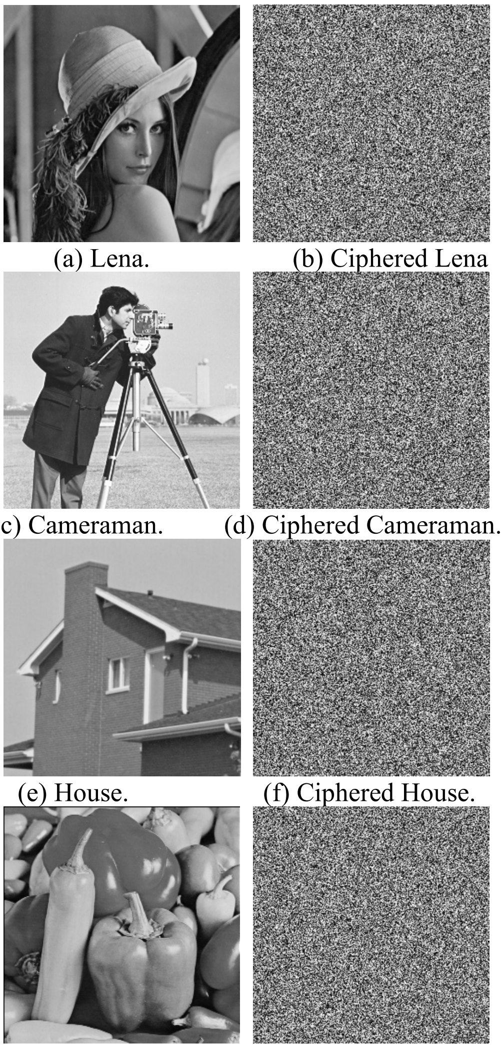
Table 3. The order of bits.

The lists	The order of bits
$Q_1^{(perm)}$	{2,9,13,1,10,8,5,12,3,14,7,15,4,0,6,11}
$Q_2^{(perm)}$	{3,14,6,15,10,8,4,12,13,11,7,1,5,0,2,9}
$Q_1^{(diff)}$	{7,1,3,6,4,5,2,0}
$Q_2^{(diff)}$	{2,5,6,0,4,3,7,1}

Fig. 4 shows the encryption results for the images and the cipher images are recovered correctly.

4. The security analysis

The security of the cryptosystem and specific example is presented to show the effectiveness.



(a) Lena. (b) Ciphered Lena

(c) Cameraman. (d) Ciphered Cameraman.

(e) House. (f) Ciphered House.

(g) Peppers. (h) Ciphered Peppers.

Fig. 4. The plain images (left column) and its ciphered images (right column).

4.1. The key space

The rule of bits to be extracted and extended in the permutation and diffusion can be considered as the contribution to the key space. As shown in this example with the value $r^{(perm)} = r^{(diff)} = 3.625$, these are represented in the format of fixed point as $11101000000000000000000000000000$. So the number of bits can be perturbed is 29 bits for each of control parameters in this example. There is a number of possible ways of extension for k_1 and k_2 bits laid out of 29 bits in the permutation and the diffusion processes. Moreover, the same number of possible ways of extraction for bits to get pixel positions XY_{new} and the value of cipher pixels C_{XY} from X_n.

The location information for the extraction and extension is considered as the encryption key of the proposed cryptosystem. That is presented by the value of elements in $Q_1^{(perm)}$, $Q_1^{(diff)}$, $Q_2^{(perm)}$, and $Q_2^{(diff)}$. The values of the bit order can be ranged from 0 to 30, so, each element is encoded by 5 bits. It is noted that each elements of $Q_1^{(perm)}$, and $Q_2^{(perm)}$ has 16 elements and each element represented by 5 bits, while each elements of $Q_1^{(diff)}$and $Q_2^{(diff)}$has 8 elements and each is encoded by 5 bits. The total number of bits represents for the order of bits N_{order} in $Q_1^{(perm)}$, $Q_1^{(diff)}$, $Q_2^{(perm)}$, and $Q_2^{(diff)}$is $N_{order} = 5 \times (2 \times 16 + 2 \times 8) = 240$ bits. As given in Table 2, the number of bits representing for parameters is $N_{para} = 138$. Therefore, the key space is $N_{Space} = N_{para} + N_{order} (= 378$ bits in this case).

It is clear that the key space $N_{Space}$is expanded more much significantly in compared with that in the case of without considering the rule of bits to be extracted and extended.

4.2. The key sensitivity analysis

The key sensitivity analysis is considered by means of the difference in ciphertexts obtained by smallest difference in encryption keys [8]. Let us call ΔK be the tolerance in the encryption key. The ciphertext difference rate (Cdr) is reflected for the key sensitivity and calculated by

$$Cdr = \frac{Diff(C,C_1)+Diff(C,C_2)}{2M\times N} \times 100\%. \qquad (9)$$

M and N are the number of pixel rows and columns; C is the ciphertext using the encryption key K; C_1 and $C_2$are ciphertexts using the encryption key $K + \Delta K$ and $K - \Delta K$, respectively. There, $Diff(A, B)$ is the number of pixels with the value in A different from that in B, i.e.

$$Diff(A,B) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} Difp(A(x,y), B(x,y)), \qquad (10)$$

where $Difp(A(x,y), B(x,y))$ is

$$Difp(A(x,y), B(x,y)) = \begin{cases} 1, \text{for } A(x,y) \neq B(x,y); \\ 0, \text{ for } A(x,y) = B(x,y). \end{cases} \qquad (11)$$

The simulation is carried out with smallest difference in every single element contributing to the encryption key, i.e. $r^{(perm)}$, $r^{(diff)}$, $IV^{(perm)}$, $IV^{(diff)}$, and C_0. The smallest amount of tolerance ΔK is defined by the resolution of the value representation, i.e. one bit of LSB. The value of other parameters is as chosen earlier. In addition, the simulation performs on a set of 100 random images with the same size of 256×256. The means and standard deviation of Cdr is as shown in Table 4.

Table 4. The key sensitivity Cdr.

On parameter s	Cdr					
	Mean (%)			Std. dev. (%)		
N_{round}	1	2	3	1	2	3
$r^{(perm)}$	99.8	99.9	99.9	0.015	0. 01	0.01
$r^{(diff)}$	99.7	99.8	99.8	0.02	0.02	0.01
$IV^{(perm)}$	98.6	98.8	99.0	0.012	0.01	0.01
$IV^{(diff)}$	98.9	98.9	99.0	0.02	0.015	0.01
C_0	99.0	99.4	99.5	0.021	0.02	0.01

It is clear from Table 4 that the rate of change is very large and it seems to need a single encryption round to get the value of almost pixels changed. The maximum value of Cdr is 99.9% at the third round is much better than that of 99.63% as the result of work of [10] published very recently.

4.3. Information entropy analysis

In this work, the well-known Shannon's information entropy is applied to measure the randomness in the data of cipher images. The information entropy (IE) is

$$IE(m) = \sum_{i=0}^{2^{k_2}-1} p(m_i) \log_2 \frac{1}{p(m_i)}, \qquad (12)$$

where k_2 is the number of bits representing for the pixel value; m_i is the value of pixels; and $p(m_i)$is the probability of m_i occcurence. In this work, four well-known grayscale images with the size of 256×256 are used for measuring IE of the cipher images, i.e. *Lena*, *Cameramen, House,* and *Peppers*. The IE with respect for a number of encryption rounds is obtained as given in Table 5. Specifically, the value of $IE = 7.999$ is almost equal to the theoritical value of 8. It is much better than that most of recent results, i.e. 7,9972 in [11], 7,9974 in [12], and 7,9965 in [9], etc.

Table 5. The IE of the ciphered image.

Name	IE		
N_{round}	1	2	3
Lena	7.990	7.990	7.998
Cameraman	7.980	7.990	7.999
House	7.970	7.989	7.999
Peppers	7.985	7.987	7.999

4.4. Differential analysis

The differential analysis is to test the cryptosysem against from differential attacks by using the slightly difference in the plaintexts with respect to that in the corresponding ciphertexts. Here, the analysis for the differential attack is based on the two well-known measures, i.e. Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI). The equation to calculate $NPCR$ and $UACI$ are as below.

$$NPCR = \frac{\sum_{x,y} D(x,y)}{M \times N} \times 100\%, \quad (13)$$

$$UACI = \frac{1}{M \times N} \frac{\sum_{x,y} |C_1(x,y) - C_2(x,y)|}{255} \times 100\%, \quad (14)$$

$$D(x,y) = \begin{cases} 1, for\ C_1(x,y) \neq C_2(x,y) \\ 0,\ for\ C_1(x,y) = C_2(x,y) \end{cases}. \quad (15)$$

There, $C_1(x,y)$and $C_2(x,y)$ are pixels at the location (x,y) in the cipher images with a slightly difference in their corresponding plain images.

In this work, a set of 100 plain images with the random value of pixels and the size of 256×256 are used. Encryption is carried out with the set of plain images P_1 to produce a set of cipher images C_1. The slightly difference is made to each of images in P to get P_2 . A modification with randomly increased or decreased by 1 to the value of randomly chosen pixel, except for the pixels with the value of 0 and 255. Then, images in P_2 are encrypted with different encryption rounds (N_{round}) to produce a set of cipher images C_2. Calculation for $NPCR$ and $UACI$ is performed on $C_1$and C_2.

The average of $NPCR$ and $UACI$ as well as the standard deviation are calculated over the set of 100 images as described. The result is shown in Table 6 that the $NPCR$ and $UACI$ of proposed cryptosystem is as good as those in the recent results published by in [9,10] and better than those cited therein.

Table 6 shows that the maximum NPCR and UACI are 99.9 and 33.451, respectively. The result of NPCR in this work is better that that of 99.6126 in [9], of 99.61 in [10], and of 99.5974 in [13], etc.

Table 6. The average of $NPCR$ and $UACI$ calculated over 100 pixels.

N_{round}	$NPCR$ (%)		$UACI$ (%)	
	Mean	Std. dev.	Mean	Std. dev.
1	99.6	0.030	33.451	0.020
2	99.8	0.012	33.450	0.015
3	99.9	0.010	33.430	0.011

5. Conclusion and future works

The paper has presented the novel cryptosystem using the Logistic map. The main contribution of this work is that the cryptosystem works on the bit level perturbation on the control parameter of the Logistic map. The key space is 378 bits and this is very large in compared with other cryptosystems. With this key space, the brute-force attack must be unsuccessful on nowadays computers. The simulation results also show that the proposed cryptosystem can be as good as those published recently.

The future work will be the investigation for the bit level perturbation on chaotic values of state variables, and for the implementation them on the reconfigurable hardware.

Acknowledgments

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.04-2018.06.

References

[1] L. Kocarev, S. Lian, Chaos-Based Cryptography: Theory, Algorithms and Applications, Springer Berlin Heidelberg (2011).

[2] L. Kocarev, Chaos-based cryptography: a brief overview, IEEE Circuits and Systems Magazine 1 (2001) 6-21.

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

[4] S.E. Assad, M. Farajallah, A new chaos-based image encryption system, Signal Processing: Image Communication 41 (2016), 144-157.

[5] L. Liu, S. Miao, A new image encryption algorithm based on logistic chaotic map with varying parameter, SpringerPlus 5 (2016) 289.

[6] W. Zhang, K-W. Wong, H. Yu, Z-L. Zhu, A symmetric color image encryption algorithm using the intrinsic features of bit distributions, Communications in Nonlinear Science and Numerical Simulation 18 (2013) 584-600.

[7] J. Chen, Z. Zhu, C. Fu, L. Zhang, Y. Zhang, An image encryption scheme using nonlinear inter-pixel computing and swapping based permutation approach, Communications in Nonlinear Science and Numerical Simulation 23 (2015) 294 – 310.

[8] S. Lian, J. Sun, Z. Wang, Security Analysis of A Chaos-based Image Encryption Algorithm, Physica A. 351 (2005) 645 - 661.

[9] P. Ping, J. Fan, Y. Mao, F. Xu, J. Gao, A chaos based image encryption scheme using digit-level permutation and block diffusion, IEEE Access. 6 (2018) 67581 - 67593.

[10] Erdem Yavuz, A novel chaotic image encryption algorithm based on content-sensitive dynamic function switching scheme, Optics & Laser Technology 114 (2019), 224-239.

[11] Y. Li, C. Wang, and H. Chen, A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation, Optics and Lasers in Engineering 90 (2017), 238 – 246.

[12] X.Wang and H. li Zhang, A color image encryption with heterogeneous bit permutation and correlated chaos, Optics Communications 342 (2015), 51– 60.

[13] A.-V. Diaconu, Circular inter-intra pixels bit-level permutation and chaos based image encryption, Information Sciences 355-356 (2016), 314 – 327.