Novel Models of Image Permutation and Diffusion Based on Perturbed Digital Chaos

Thang Manh Hoang, Safwan El Assad

entropy

MDPI

Article

Novel Models of Image Permutation and Diffusion Based on Perturbed Digital Chaos

Thang Manh Hoang [1,*] and Safwan El Assad [2]

[1] School of Electronics and Telecommunications, Hanoi University of Science and Technology, 1 Dai Co Viet, Hai Ba Trung, Hanoi 100000, Vietnam

[2] IETR (Institut d'Electronique et des Télécommunications de Rennes), Université de Nantes, CNRS, UMR 6164, Polytech Nantes, Rue Christian Pauc CS 50609, CEDEX 3, 44306 Nantes, France; safwan.elassad@univ-nantes.fr

* Correspondence: thang.hoangmanh@hust.edu.vn; Tel.: +84-98-880-2694

Received: 12 April 2020; Accepted: 9 May 2020; Published: 13 May 2020

Abstract: Most of chaos-based cryptosystems utilize stationary dynamics of chaos for the permutation and diffusion, and many of those are successfully attacked. In this paper, novel models of the image permutation and diffusion are proposed, in which chaotic map is perturbed at bit level on state variables, on control parameters or on both. Amounts of perturbation are initially the coordinate of pixels in the permutation, the value of ciphered word in the diffusion, and then a value extracted from state variables in every iteration. Under the persistent perturbation, dynamics of chaotic map is nonstationary and dependent on the image content. The simulation results and analyses demonstrate the effectiveness of the proposed models by means of the good statistical properties of transformed image obtained after just only a single round.

Keywords: chaos-based image encryption; chaotic cryptography; dynamics perturbation; chaotic permutation; chaotic diffusion

1. Introduction

For recent decades, chaos has been discovered in natural, human, and engineering models [1]. It has been also generated by human for pragmatic applications. Two of prominent applications are chaotic communications [2] and chaos-based cryptography [3]. Recently, chaos-based image encryption has attracted increasing interest [4–7]. That is due to the good cryptographic properties of chaotic sequences [8–11] and a chaotic system can be implemented on digital hardware [12–15]. In digital hardware, dynamics of any chaotic system is degraded to periodic orbits due to the round-off errors by the limited number of bits represented for values of state variables and control parameters [16–18]. The larger the number of bits representing for chaotic state variable and control parameters is, the longer the length of period is obtained. Beside that, the period of orbits produced by a chaotic map can be lengthened by several methods as suggested in Reference [19]. Two of such methods are perturbation on chaotic states by another chaotic map [20,21] and by using linear feedback shift register (LFSR) [22].

For chaos-based cryptography, at least one of encryption processes is involved by chaos. Along with the Feistel structure, the substitution-permutation network (SPN) structure attains the properties of confusion and diffusion [23], which are widely employed in both conventional block ciphers [24,25] and chaotic ones [26–29]. Typically, the SPN structure can be realized in chaotic ciphers by means of the combination of permutation and diffusion processes, for example, References [30,31]. The advantage of the SPN structure is that the cryptographic statistics can be increased by means of increasing the number of rounds in each of permutation and diffusion processes and/or in a whole.

For most of chaos-based image cryptosystems, a chaotic system is used for generating chaotic sequences for the permutation and diffusion processes. Firstly, the chaotic permutation is implemented with the involvement of at least one chaotic system to shuffle pixels or bits of pixels within the image. The permutation rule can be static in the form of table or dynamic by inducing from chaotic values. Secondly, the chaotic diffusion is usually realized by a mixture between chaotic values and values of plain pixels. In literature, most of successful attacks on chaotic ciphers are based on weaknesses in algorithms of permutation and diffusion processes, for example, References [32–37]. Besides, the works [38,39] points out the criteria and assessment to a chaotic cryptosystem.

Under a cryptographic point of view, it is obviously that the more complicated dynamics of chaos allows the stronger chaos-based cryptosystem. Recently, many chaos-based cryptosystems were proposed with the use of more complicated chaos. Along with the use of hyperchaotic, time-delay, fractional order, and spatiotemporal chaotic systems, complicated dynamics can be obtained by mixed of various chaotic systems such as References [40–45]. In such the chaos-based cryptosystems, chaotic systems work with fixed values of control parameters and with non-disturbed chaotic orbits. In other words, dynamics of chaotic maps is stationary in generating encryption keys for the permutation and diffusion.

It is also well-known that analysis of chaotic dynamics can be performed by the observation and measurement of dynamics like trace formulas [46–49] or inference of control parameters [50–52], and so forth. Many analysis methods success with additive perturbation [46–48]. With the development of analysis methods, analysis of chaotic dynamics can used as a powerful tool to attack chaos-based cryptosystems [53]. However, to date, there has not been any report about a successful attack to a chaotic block cipher by means of analysis of chaotic dynamics. By applying analysis of chaotic dynamics, chaotic cryptosystems based on stationary dynamics will become possibly insecure in the future. Therefore, one of potential approaches of chaotic image encryption is based on perturbed chaos.

Definitely, a chaos-based cryptosystem becomes much stronger if its encryption keys are dependent on the image content. The involvement of image content in chaotic dynamics is created by an external perturbation. In fact, there are two approaches to create the connection between the image content and encryption keys, dependent on whether the image content involves in chaotic dynamics or not. Firstly, the connection between the image content and encryption keys is established by means of state perturbation, for example, References [9,42,54–58]. References [9,54] present the a selection mechanism in which the image content is used for selecting one of chaotic sequences to generate keystreams. The initial values of chaotic system are fixed, and neither state variables nor control parameters of chaotic system is disturbed during generation of chaotic sequences. As presented in Reference [42], the initial value and the value of parameter of chaotic system are generated with the use of image content for whole encryption, but the parameter of chaotic map deciding the manner of the permutation and diffusion is dependent on the image content of blocks. The advantage is that the value of parameter of chaotic system is updated after every block of image. As presented in Reference [55], the initial value of chaotic map in the diffusion process is calculated by the value of pixels, and the output of chaotic map is used to compute the ciphertext. The important point is that the image content involves in the diffusion by means of its initial value of chaotic map. Whereas, the value of control parameters is kept constant. The same approach as given in References [55,59] is used in the work by G. Ye et al. [56], in which the initial value of chaotic map is computed by information entropy of plain image. In Reference [59], the diffusion process utilizes one of state variables of hyper-chaotic Chen's system, in which only initial value of chaotic map is being updated after every pixel. In the work of H. Li et al. [58], the orbit of two-dimensional logistic-adjusted-sine map (2D-LASM) is disturbed by the coordinate and the value of pixels during the generation of the keystreams for the permutation and diffusion, while the value of control parameters of (2D-LASM) is kept constant. In another way, the initial value of chaotic map is the output of authentication by SHA-256 as in Reference [57], or the value of control parameter of chaotic map is calculated by the image content as in Reference [60] for generation of finite state machines for the diffusion. As reported in Reference [61], the value of control

parameter of Logistic map is calculated by the image content, and it is kept constant in the encryption process. The common point in those works is that control parameters of chaotic maps are unperturbed, so dynamics of chaotic maps is stationary.

Secondly, the dependence of encryption keys on the image content can be created by perturbing on control parameters of chaotic systems. To the best of our knowledge, there are a limited number of published works in this way as in References [62,63]. Specifically, in the work proposed by J. Chen et al. [62], the control parameter of Logistic map is perturbed in the pixel swapping confusion and diffusion processes. In the work by T. Song [63], the control parameter of Logistic map is computed by using the value of pixels, and updated in the diffusion process. In fact, the disadvantage is that the value range of control parameter must be always monitored and adjusted under a condition. Moreover, a number of additional arithmetic operations along with those of chaotic map requires higher computational complexity and resource.

In addition to two main approaches as above described, some other works presents the utilization of perturbed chaos for cryptosystems, for example, References [21,62,64,65]. In References [21,64,65], the chaotic maps are perturbed by additional transformations of state variables or by some conditions, rather than by information of pixels. In fact, additional equations and conditions make chaotic maps more mathematically complicated, but not really perturbed by any external force. Thus, dynamics of chaotic maps is stationary, and the vulnerability still exists [66].

Overall, perturbed dynamics of chaos with the dependence on the image content offers the cryptographic properties better than those with stationary dynamics in terms of statistics and it can resist from the type of chosen plaintext attack. However, reported image cryptosystems based on perturbed chaos have the proprietary structures with the use of specific chaotic systems, and perturbation is realized in arithmetic operations. Under the viewpoint of hardware, more arithmetic operations will require more resource and may reduce the speed of the encryption. In other words, there is a lack of models with a simpler perturbation than those in previous works, utilization of various chaotic maps, and suitability for hardware implementation.

In this paper, novel models of perturbed digital chaos are proposed for the image permutation and diffusion. Perturbation on chaotic dynamics is carried out at bit level by three schemes, that is, perturbation on state variables, on control parameters and on both state variables and control parameters ("on both" for short). Amounts of perturbation can be either the coordinate of pixels in the permutation and the value of pixels in the diffusion or the value extracted from state variables. Chaotic dynamics becomes nonstationary and it provides cryptographic advantages for the image permutation and diffusion. The example and simulation results demonstrate the effectiveness of the proposed models with the use of Logistic map. It is noted that this work will not go into analysis of dynamic properties of chaotic systems under perturbation, and ones can find that in other works, for example, References [19,46,67,68], and so forth.

The main contributions of the work are as follows: The structures of chaotic perturbation with an external force are generalized, in which three schemes of perturbation are clearly expressed. The models of the permutation and diffusion for the chaotic image encryption are proposed by means of utilizing the corresponding schemes of perturbation. The perturbation is the coordinate of pixels in the permutation and the value of pixels in the diffusion. The statistical and security analyses are carried out for the example using the generic Logistic map as a proof of effectiveness of the proposed models.

The rest of the paper is organized as follows—Section 2 presents some basic preliminaries. The general structures of perturbed chaotic map are given in Section 3. Next, the proposed models of permutation and diffusion for chaotic image encryption are detailed in Section 4. Section 5 shows the example and the simulation results for the permutation and diffusion with various schemes of perturbation using the Logistic map. Finally, Section 6 gives some concluding remarks of the work.

2. Some Basic Preliminaries

2.1. Representation of Images

Let us consider the raster format of a grayscale image is represented as a 2-dimensional matrix I with the size $M \times N$. The element of I at the location (x, y) is called a pixel $P(x, y)$ in binary of k bits as $P_{XY} = b_{k-1}b_{k-2}...b_1b_0$. The image can be considered as a collection of pixels represented by

$$I = \bigcup_{x=0}^{M-1} \bigcup_{y=0}^{N-1} P(x, y), \tag{1}$$

where M and N are the number of rows and columns of pixels, respectively. In the following text, an entity formed by a collection of elements is denoted by $\bigcup$ with an associated index. In the case of RGB image, each of three color layers can be considered as a grayscale image.

2.2. Bit Representation for Real Numbers

Let us consider that a chaotic map in Equation (6) is implemented in a digital platform. So, the value of state variables and that of control parameters are represented in one of two formats, that is, fixed-point or floating-point number. Fixed-point representation is suitable for most chaotic maps because the value of state variables and of control parameters are in the narrow ranges. Signed and unsigned fixed-point numbers are illustrated in Figure 1.

Integer	Fraction
$m^{(int)}$	$m^{(frac)}$

(**a**) Unsigned fixed-point

Sign	Integer	Fraction
1	$m^{(int)}$	$m^{(frac)}$

(**b**) Signed fixed-point

Figure 1. Representation of fixed-point number.

For example, the Logistic map has the range of (0,1) for chaotic state variable, and that of $[3.57, 4.0]$ for the control parameter, thus, the format of unsigned fixed-point is suitable. The number of bits required for the integer part in the value of chaotic state and of control parameter are 1 and 3, respectively.

For a signed fixed-point representation, a real number is represented one bit for the sign S, $m^{(int)}$ bits for the integer part and $m^{(frac)}$ bits for the fractional part; or $m = 1 + m^{(int)} + m^{(frac)}$. The fixed-point number can be written in sequence of bits as $(S)b_{m^{(int)}-1}...b_0(.)b_{-1}...b_{-m^{(frac)}}$. Note that the binary point is in the parentheses, '(' and ')'. The value is $V = (-1)^S \sum_{i=-m^{(frac)}}^{m^{(int)}-1} b_i \times 2^i$.

For a unsigned fixed-point representation, there is no sign bit. Thus, the number of bits is $m = m^{(int)} + m^{(frac)}$; representation in bit sequence is $b_{m^{(int)}-1}...b_0(.)b_{-1}...b_{-m^{(frac)}}$; and the value is $V = \sum_{i=-m^{(frac)}}^{m^{(int)}-1} b_i \times 2^i$.

As a real number is represented as a bit sequence, bitwise operations can be applied to change the state of bits.

2.3. Representation of Bit Sequence and Bit Arrangement

Here, the bit arrangement is to permute bits, but the term "bit arrangement" is used to avoid confusing the "permutation" of pixels in the later part of the paper. Let us consider two arrays of bit sequences $A = (A_i)_{1 \le i \le I_A}$ and $B = (B_i)_{1 \le i \le I_B}$. Bit sequences of A and B are $A_i = \bigcup_{j=1}^{J_A} a_{i,j}$ and $B_i = \bigcup_{j=1}^{J_B} b_{i,j}$, respectively. There, $a_{i,j}$ and $b_{i,j}$ are j^{th} bits of i^{th} sequences, and J_A and J_B are the lengths of bit sequences A_i and B_i, respectively. In order to simplify for the representation, the size of A and B is denoted by $I_A \times J_A$ and $I_B \times J_B$, respectively.

Let us define a bit arrangement for a general case of $I_A \neq I_B$ and $J_A \neq J_B$. Bit sequences of A are constructed by bits from sequences of B. The rule of bit arrangement is encoded by a matrix Y, and the arrangement operator is denoted by $\circ$, such that $A = Y \circ B$. For the matrix $Y = \left[y_{i,j}\right]_{1 \leq i \leq I_A,\, 1 \leq j \leq J_A}$, $y_{i,j}$ is the combination of indexes indicating a bit of B. Each row of Y, $Y_i = \left[y_{i,j}\right]_{1 \leq j \leq J_A}$, is used for constructing a bit sequence A_i, in other words, bit sequences of A are $A_i = Y_i \circ B$. It is noted that a bit $b_{i,j}$ of B can be used multiple times in A.

For example, the array of bit sequences B has the size of $(I_B, J_B) = (5, 4)$ as

$$B = \begin{pmatrix} b_{1,1}b_{1,2}b_{1,3}b_{1,4} \\ b_{2,1}b_{2,2}b_{2,3}b_{2,4} \\ b_{3,1}b_{3,2}b_{3,3}b_{3,4} \\ b_{4,1}b_{4,2}b_{4,3}b_{4,4} \\ b_{5,1}b_{5,2}b_{5,3}b_{5,4} \end{pmatrix}. \tag{2}$$

The array A is constructed from bits of B. A is with three bit sequences, and each sequence has six bits; or the size of A is $(I_A, J_A) = (3, 6)$. The matrix Y is

$$Y = \begin{bmatrix} (4,3) & (1,4) & (2,3) & (4,1) & (2,2) & (3,1) \\ (1,3) & (3,4) & (3,2) & (2,1) & (5,4) & (3,2) \\ (4,3) & (3,2) & (1,4) & (3,1) & (5,3) & (2,2) \end{bmatrix}. \tag{3}$$

So, the array A is as

$$A = Y \circ B = \begin{pmatrix} b_{4,3}b_{1,4}b_{2,3}b_{4,1}b_{2,2}b_{3,1} \\ b_{1,3}b_{3,4}b_{3,2}b_{2,1}b_{5,4}b_{3,2} \\ b_{4,3}b_{3,2}b_{1,4}b_{3,1}b_{5,3}b_{2,2} \end{pmatrix}, \tag{4}$$

where, three bit sequences are $A_1 = b_{4,3}b_{1,4}b_{2,3}b_{4,1}b_{2,2}b_{3,1}$, $A_2 = b_{1,3}b_{3,4}b_{3,2}b_{2,1}b_{5,4}b_{3,2}$ and $A_3 = b_{4,3}b_{3,2}b_{1,4}b_{3,1}b_{5,3}b_{2,2}$.

If a certain bit of A_i is fixed with a predefined state '0' or '1', the terms BIT_0 and BIT_1 are used to indicate the states '0' and '1' in Y_i, respectively. For instance, the value of bits in A is fixed such as $A_1 = (1)b_{1,4}b_{2,3}b_{4,1}b_{2,2}(0)$, so Y_1 must be as $Y_1 = [BIT_1, (1,4), (2,3), (4,1), (2,2), BIT_0]$.

Moreover, except for a number of bits with fixed values, let us define H_Y to be the number of bits in A taken from B as

$$H_Y = \sum_{i=1}^{I_A} \sum_{j=1}^{J_A} y_{i,j} \qquad \forall y_{i,j} \notin (BIT_0, BIT_1). \tag{5}$$

These will be used in the proposed models of permutation and diffusion in the later part of the paper.

3. Perturbed Digital Chaotic Map

It is definitely that dynamics of a chaotic system becomes nonstationary if the chaotic system is perturbed by an external force. In this section, a chaotic map with perturbation at bit level is described in two primitive schemes, that is, perturbation on state variables and on control parameters of chaotic map. The third scheme is the combination of two mentioned ones, in which both state variables and control parameters of chaotic map are perturbed. In any scheme, dynamics of chaotic system becomes complicated and that brings advantages in terms of cryptographic properties.

Let us consider a chaotic map defined by

$$\begin{cases} X_{n+1} = F(X_n, \Gamma), \\ X_n = [x_n^{(D)} \;\; x_n^{(D-1)} \ldots x_n^{(2)} \;\; x_n^{(1)}], \\ \Gamma_n = [\gamma_n^{(G)} \;\; \gamma_n^{(G-1)} \ldots \gamma_n^{(2)} \;\; \gamma_n^{(1)}], \end{cases} \tag{6}$$

where X_n and Γ_n are vectors of chaotic state variables and of control parameters, respectively; D is the number of dimensions, and G is the number of control parameters; $D = ||X_n||$ and $G = ||\Gamma_n||$. The perturbation on state variables is

$$X_{n+1} = F(\hat{X}_n, \Gamma_0), \tag{7}$$

on control parameters is

$$X_{n+1} = F(X_n, \hat{\Gamma}_n), \tag{8}$$

and on the both of state variables and control parameters is

$$X_{n+1} = F(\hat{X}_n, \hat{\Gamma}_n). \tag{9}$$

There, $\hat{X}_n$ and $\hat{\Gamma}_n$ are the perturbed variables and control parameters, respectively described as

$$\begin{cases} \hat{X}_n &= \Psi_X(X_n, \Delta_X), \\ \Delta_X &= \Omega_X(X_n, E_X), \\ \Delta_X &= [\delta_X^{(D)} \ \ \delta_X^{(D-1)} ... \ \delta_X^{(2)} \ \ \delta_X^{(1)}]^\intercal, \end{cases} \tag{10}$$

and

$$\begin{cases} \hat{\Gamma}_n &= \Psi_\Gamma(\Gamma_n, \Delta_\Gamma), \\ \Delta_\Gamma &= \Omega_\Gamma(X_n, E_\Gamma), \\ \Delta_\Gamma &= [\delta_\Gamma^{(G)} \ \ \delta_\Gamma^{(G-1)} ... \ \delta_\Gamma^{(2)} \ \ \delta_\Gamma^{(1)}]^\intercal. \end{cases} \tag{11}$$

There, $\hat{X}_n$ and $\hat{\Gamma}_n$ are $\hat{X}_n = [\hat{x}_n^{(D)} \ \ \hat{x}_n^{(D-1)} ... \ \hat{x}_n^{(2)} \ \ \hat{x}_n^{(1)}]$ and $\hat{\Gamma}_n = [\hat{\gamma}_n^{(G)} \ \hat{\gamma}_n^{(G-1)} ... \ \hat{\gamma}_n^{(2)} \ \hat{\gamma}_n^{(1)}]$, respectively; Δ_X and Δ_Γ are instant amounts of perturbation; Ψ_X and Ψ_Γ define the operations of perturbation; $\Omega_X = \{\omega_X^{(i)}, i = \{1, ..., D\}\}$ and $\Omega_\Gamma = \{\omega_\Gamma^{(i)}, i = \{1, ..., G\}\}$ are sets of functions to produce amounts of perturbation; E_X and E_Γ are vectors of external forces. Note that, the subscripts X and Γ denote for the notations belonging to state variables and control parameters, respectively.

In hardware perspective, values of state variables and control parameters are represented in a format of real number. Thus, all of functions, that is, Ω_X, Ω_Γ, Ψ_X, and Ψ_Γ, operate at bit level. Specifically, at bit level, each bit of operands in such the functions can be manipulated by basic logic gates AND, OR, NOR, NAND, NOT, XOR, XNOR or their combination.

Figure 2 illustrates the proposed schemes of perturbation. The perturbation can be on chaotic state variables, control parameters, or both. IV is a vector of initial condition.

In any scheme, perturbation must ensure that chaos exhibits and values of X_n and Γ_n must be within valid ranges. As given in Equations (10) and (11), the value ranges of X_n and Γ_n are dependent on both amounts and functions of perturbation. At bit level, values of X_n and Γ_n are represented in the format of fixed point as shown in Section 2.2. Therefore, Δ_X and Δ_Γ define the perturbation to X_n and Γ_n, respectively. The disturbance level on a chaotic map is really dependent on the position of perturbed bits in the representation.

Also, values of $\hat{X}_n$ and $\hat{\Gamma}_n$ are represented by a number of bits, and specific position of perturbed bits is pointed out by perturbation functions $\Psi_X(.)$ and $\Psi_\Gamma(.)$. Let us Θ_X and Θ_Γ respectively be vectors of value tolerances of state variables and control parameters, $\Theta_X = |\hat{X}_n - X_n|$ and $\Theta_\Gamma = |\hat{\Gamma}_n - \Gamma_n|$. Equivalently, at each time of perturbation, $\hat{X}_n$ and $\hat{\Gamma}_n$ in Equations (7)–(9) are

$$\hat{X}_n = X_n \pm \Theta_X, \tag{12}$$

and,

$$\hat{\Gamma}_n = \Gamma_n \pm \Theta_\Gamma, \tag{13}$$

where, $\Theta_X = \{\theta_X^{(i)}, i = \{1, ..., D\}\}$ and $\Theta_\Gamma = \{\theta_\Gamma^{(i)}, i = \{1, ..., G\}\}$. As described above, values of $\theta_X^{(i)}$ and $\theta_\Gamma^{(i)}$ are dependent on the state of bits in $\delta_X^{(i)}$ and $\delta_\Gamma^{(i)}$ making bits in $x_n^{(i)}$ and $\gamma_n^{(i)}$ changed. The value

ranges of $\theta_X^{(i)}$ and $\theta_\Gamma^{(i)}$ can be figured out when the positions of perturbed bits are known in a specific scheme of perturbation. In general, this can be defined by ones, and the larger amounts of perturbation will make the more complexity in chaotic dynamics. This suggests that the higher significant bits of X_n and Γ_n should be perturbed.

As shown in Equations (10) and (11), the amounts of perturbation, Δ_X and Δ_Γ, are dependent on pairs of values (X_n,E_X) and (X_n,E_Γ), respectively. At bit level, all functions of Ω_X and Ω_Γ are bitwise operations, thus basic logic gates and their combination can be used for bit manipulation.

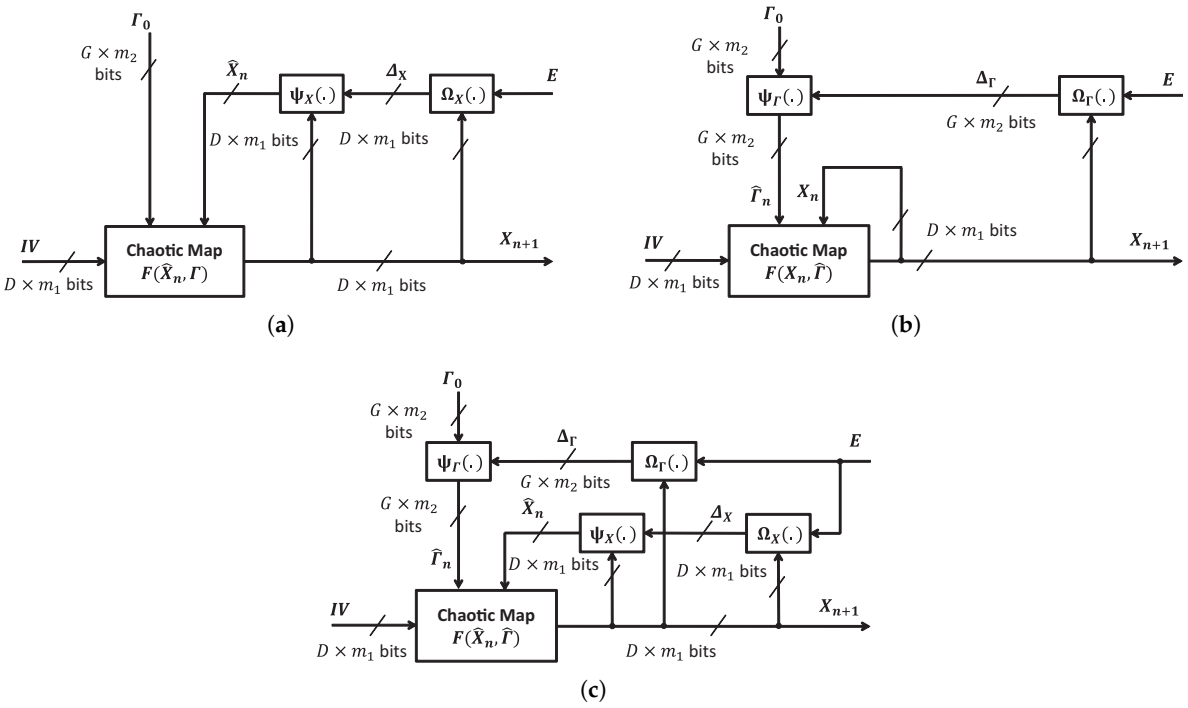


Figure 2. Chaotic map with perturbations (**a**) on state variables, (**b**) control parameters, and (**c**) both of state variables and control parameters.

4. Proposed Models of Permutation and Diffusion

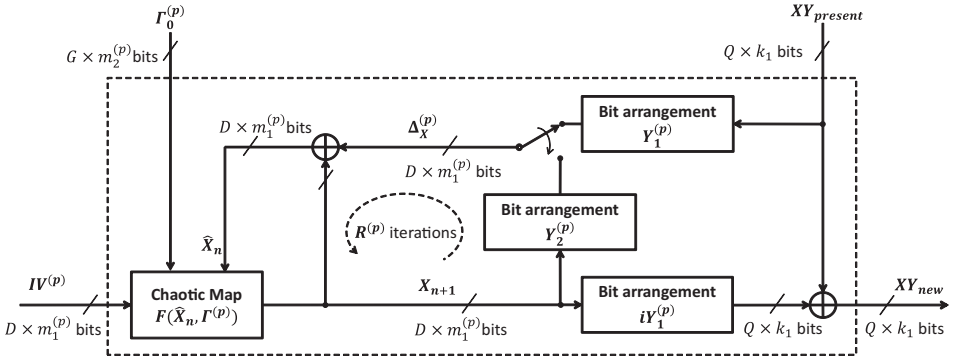
In this section, the models of permutation and diffusion are proposed those are based on the proposed schemes of perturbation as described in the previous Section. It is noted that the *XOR* operation is chosen as the function of perturbation. The superscripts (p) and (d) are associated on the notations to indicate the permutation and diffusion.

4.1. Proposed Chaotic Pixel Permutation (CPP) with Perturbation

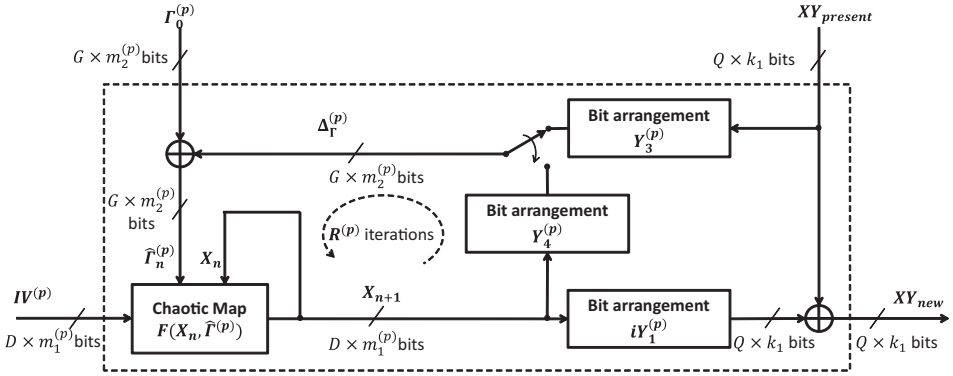
Pixel permutation shuffles pixels within the space of image using chaos. The idea of bit-level perturbation to chaotic map as illustrated in Section 3 is employed to propose three configurations of CPP as illustrated in Figure 3. The perturbation to a chaotic system is carried out on state variables (CPP-1), on control parameters (CPP-2), and on both (CPP-3) as in Figure 3a, Figure 3b, and Figure 3c, respectively.

It is assumed that a D-dimensional chaotic map $F(.)$ has G control parameters. Values of state variables and control parameter are represented in the fixed-point format by $m_1^{(p)}$ and $m_2^{(p)}$ bits, respectively. So, values of state variables X_n and its perturbation $\Delta_X^{(p)}$ can be seen as arrays of bit sequences with the size of $D \times m_1^{(p)}$. Similarly, values of control parameters $\Gamma_n^{(p)}$ and its perturbations $\Delta_\Gamma^{(p)}$ are represented by arrays of bit sequences with the size of $G \times m_2^{(p)}$ bits. Bit arrangements $Y_1^{(p)}$,

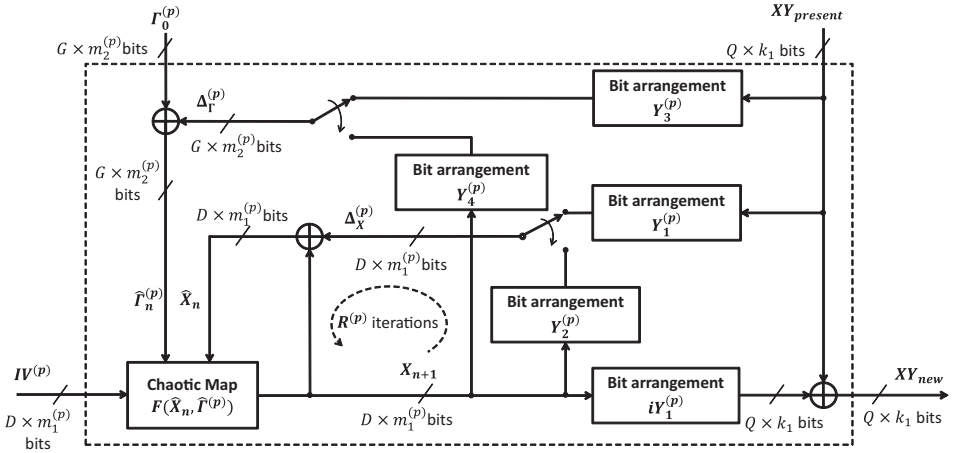
$iY_1^{(p)}$, $Y_2^{(p)}$, $Y_3^{(p)}$, and $Y_4^{(p)}$ are to arrange the size of arrays of bit sequences as described Section 2.3. The size of inputs and outputs is given in Table 1.



(**a**) Chaotic pixel permutation with the perturbation on state variables (CPP-1)



(**b**) Chaotic pixel permutation with the perturbation on control parameters (CPP-2)



(**c**) Chaotic pixel permutation with the perturbation on both (CPP-3)

Figure 3. The structure of Chaotic Pixel Permutations (CPPs) with the perturbation.

In chaotic behavior, there are constraints in the value ranges of chaotic state variables and control parameters. Specifically, the constraints are met by fixing a number of bits in chaotic state variables and in control parameters, while the rest number of bits can be changeable by the perturbation. So, the number of bits $Q \times m_1^{(p)}$ representing for the coordinate of pixels $XY_{present}$ and XY_{new} must be less than the number of changeable bits in all the schemes of perturbation. For simplest case,

the coordinate of pixels is encoded by a sequence of k_1 bits, in which row and column numbers of pixels are respectively represented by $k_1^{(x)}$ and $k_1^{(y)}$ bits; $k_1 = k_1^{(x)} + k_1^{(y)}$.

The *XOR* operation is chosen as the perturbation functions Ψ_X and Ψ_Γ in Equations (10) and (11). In this paper, bit arrangements play a role of the sets of functions $\Omega_X(.)$ and $\Omega_\Gamma(.)$ generating amounts of perturbation $\Delta_X^{(p)}$ and $\Delta_\Gamma^{(p)}$, respectively. For the CPP-1, the chaotic map is perturbed by means of modification of bits in chaotic state variables X_n with bits in an amount of perturbation $\Delta_X^{(p)}$ after every iteration n ($1 \leq n \leq R^{(p)}$), while the value of control parameters $\Gamma^{(p)}$ is kept constant. Therefore, the deterministic orbit of chaotic map is destroyed by such the perturbation amount $\Delta_X^{(p)}$. Similarly, the value of control parameters of chaotic map $\Gamma^{(p)}$ are changed after every iteration in the CPP-2. Under the perturbation on control parameters, dynamics of chaotic map becomes nonstationary. The CPP-3 is the combination of the CPP-1 and CPP-2 that both state variables and control parameters are updated after every iteration.

Table 1. Bit arrangement.

Bit Arrangements	Size of Inputs	Size of Outputs
$Y_1^{(p)}$	$Q \times k_1$	$D \times m_1^{(p)}$
$Y_3^{(p)}$		$G \times m_2^{(p)}$
$iY_1^{(p)}$	$D \times m_1^{(p)}$	$Q \times k_1$
$Y_2^{(p)}$		$D \times m_1^{(p)}$
$Y_4^{(p)}$		$G \times m_2^{(p)}$

Respectively, the state variables of chaotic map with the perturbation as given in Figure 3a–c are

$$\begin{cases} \hat{X}_0 = IV^{(p)} \oplus \Delta_X^{(p)}, \\ X_{n+1} = F(\hat{X}_n, \Gamma_0^{(p)}) \quad \text{for } n = \{1...R^{(p)}\}, \end{cases} \tag{14}$$

$$\begin{cases} X_0 = IV^{(p)}, \\ \hat{\Gamma}_0^{(p)} = \Gamma_0^{(p)} \oplus \Delta_\Gamma^{(p)}, \\ X_{n+1} = F(X_n, \hat{\Gamma}_n^{(p)}) \quad \text{for } n = \{1...R^{(p)}\}, \end{cases} \tag{15}$$

$$\begin{cases} \hat{X}_0 = IV^{(p)} \oplus \Delta_X^{(p)}, \\ \hat{\Gamma}_0^{(p)} = \Gamma_0^{(p)} \oplus \Delta_\Gamma^{(p)}, \\ X_{n+1} = F(\hat{X}_n, \hat{\Gamma}_n^{(p)}) \quad \text{for } n = \{1...R^{(p)}\}. \end{cases} \tag{16}$$

The perturbed state variables and control parameters are as

$$\begin{aligned} \hat{X}_n &= X_n \oplus \Delta_X^{(p)}, \\ \hat{\Gamma}_n^{(p)} &= \Gamma_n^{(p)} \oplus \Delta_\Gamma^{(p)}. \end{aligned} \tag{17}$$

Amounts of perturbation are represented in arrays of bit sequences $\Delta_X^{(p)}$ and $\Delta_\Gamma^{(p)}$ after bit arrangements as

$$\Delta_X^{(p)} = \begin{cases} Y_1^{(p)} \circ XY_{present} & \text{for } n = 1; \\ Y_2^{(p)} \circ X_n & \text{for } 2 \leq n \leq R^{(p)}, \end{cases} \tag{18}$$

and

$$\Delta_\Gamma^{(p)} = \begin{cases} Y_3^{(p)} \circ XY_{present} & \text{for } n = 1; \\ Y_4^{(p)} \circ X_n & \text{for } 2 \leq n \leq R^{(p)}, \end{cases} \tag{19}$$

After $R^{(p)}$ iterations, the value of $X_{R^{(p)}}$ is used to obtain the new coordinate of pixels as

$$XY_{new} = XY_{present} \oplus (iY_1^{(p)} \circ X_{R^{(p)}}). \tag{20}$$

It is noted that $\circ$ is the bit arrangement as given in Section 2.3.

4.2. Inverse Chaotic Pixel Permutation

Let us consider Inverse Chaotic Pixel Permutation (iCPP) as shown in Figure 3. The present coordinate of pixels is converted into bits sequence $XY_{present}$, and the XOR operation is used to produce new position, XY_{new}, at the last iteration. Therefore, corresponding to CPP in Figure 3, there are three structures of iCPP which are denoted iCPP-1, iCPP-2, and iCPP-3, dependent on the way of perturbation to the chaotic map. The structure of iCPP-1, iCPP-2, and iCPP-3 is identical to that of CPP-1, CPP-2, and CPP-3, respectively, as illustrated in Figure 3. The equations describing for iCPPs are the same those for CPPs as in Equations(14)–(20). The value of all parameters in iCPPs must be set the same as that in the corresponding CPPs to recover the original position of pixels as explained in Section 4.1. The main difference between iCPPs and CPPs is that pixels of image in iCPPs are permuted in a reverse direction in compared with that in CPPs of the encryptor, for example, from the pixel at position $(M-1, N-1)$ backward to $(0, 0)$.

4.3. Chaotic Diffusion with Perturbation

The idea of bit-level perturbation to chaotic map as illustrated in Section 3 is again employed to propose three configurations of chaotic diffusion (CD) in Figure 4. The chaotic system is perturbed on state variables, control parameters and on both as illustrated in Figure 4a, Figure 4b, and Figure 4c, respectively. Here, the difference in these structures in compared with those of CPPs is the feedback of C_{XY}. Pixels are diffused sequentially. Array of bit sequences C_0 with the size of $Z \times k_2$ as an initial ciphertext is used for the first pixel of diffusion. P_{XY} and C_{XY} with the size of $Z \times k_2$ are arrays of bit sequences of plaintext and ciphertext, respectively. The bit arrangements in the diffusion $Y_1^{(d)}$, $iY_1^{(d)}$, $Y_2^{(d)}$, $Y_3^{(d)}$, $Y_4^{(d)}$, $Y_5^{(d)}$ and $iY_5^{(d)}$, are with the size of inputs and outputs as shown in Table 2. Notably, the constraint is that $Z \times k_2$ must be less than the number of changeable bits of state variables and control parameters in all schemes of perturbation. As a simplest application, the value of pixels is represented by a sequence of k_2 bits.

Table 2. Bit arrangement.

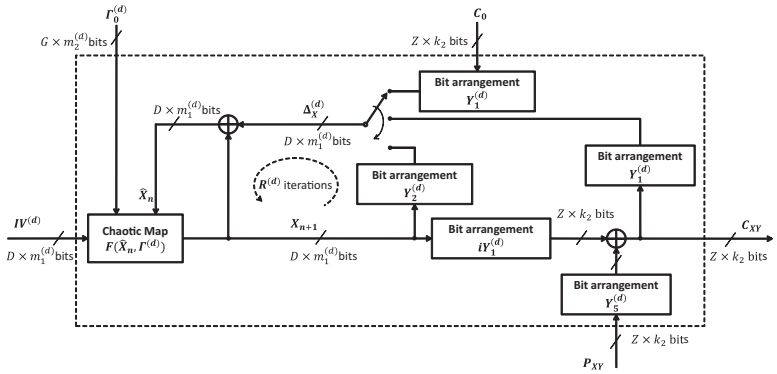
Bit Arrangements	Size of Inputs	Size of Outputs
$Y_1^{(d)}$	$Z \times k_1$	$D \times m_1^{(d)}$
$Y_3^{(d)}$		$G \times m_2^{(d)}$
$iY_1^{(d)}$	$D \times m_1^{(d)}$	$Q \times k_1$
$Y_2^{(d)}$		$D \times m_1^{(d)}$
$Y_4^{(d)}$		$G \times m_2^{(d)}$
$Y_5^{(d)}$	$Z \times k_2$	$Z \times k_2$
$iY_5^{(d)}$		

Respectively, three equations describing the diffusion as displayed in Figure 4a, Figure 4b, and Figure 4c are

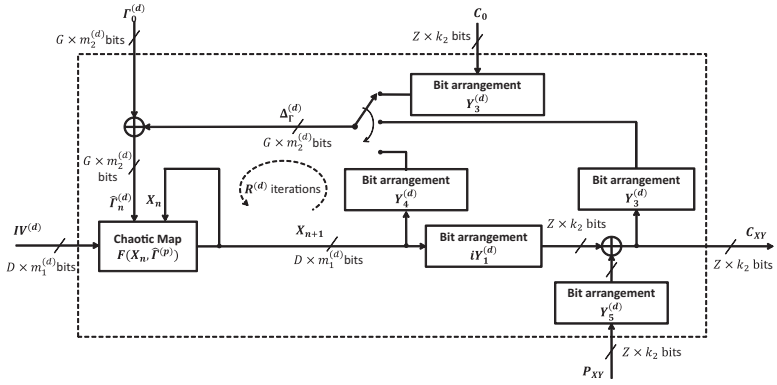
$$\begin{cases} \hat{X}_0 = IV^{(d)} \oplus \Delta_X^{(p)}, \\ X_{n+1} = F(\hat{X}_n, \Gamma_0^{(d)}) \quad \text{for } n = \{1...R^{(d)}\}, \end{cases} \tag{21}$$

$$\begin{cases} X_0 = IV^{(d)}, \\ \hat{\Gamma}_0^{(p)} = \Gamma_0^{(p)} \oplus \Delta_\Gamma^{(p)}, \\ X_{n+1} = F(X_n, \hat{\Gamma}_n^{(d)}) \quad \text{for } n = \{1...R^{(d)}\}, \end{cases} \tag{22}$$

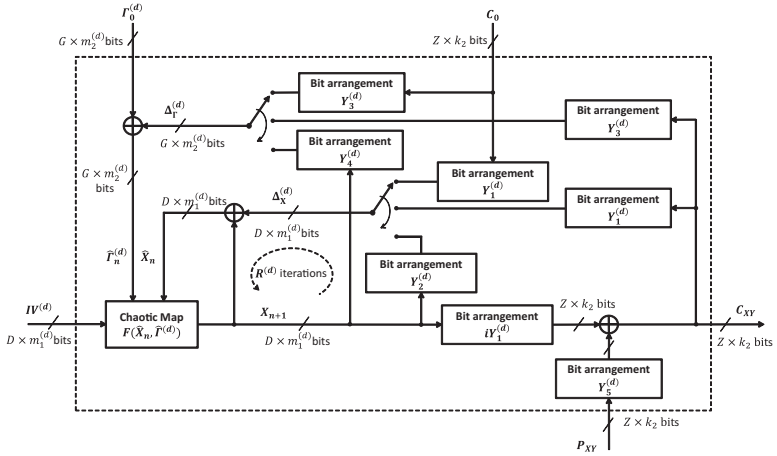
$$\begin{cases} \hat{X}_0 = IV^{(d)} \oplus \Delta_X^{(p)}, \\ \hat{\Gamma}_0^{(p)} = \Gamma_0^{(p)} \oplus \Delta_\Gamma^{(p)}, \\ X_{n+1} = F(\hat{X}_n, \hat{\Gamma}_n^{(d)}) \quad \text{for } n = \{1...R^{(d)}\}. \end{cases} \tag{23}$$



(**a**) Chaotic diffusion with the perturbation on chaotic state variable (CD-1)



(**b**) Chaotic diffusion with the perturbation on control parameter (CD-2)



(**c**) Chaotic diffusion with the perturbation on both (CD-3)

Figure 4. The structure of chaotic diffusions (CDs) with the perturbation.

The perturbed state variables and control parameters in Equations (21)–(23) are

$$\begin{aligned} \hat{X}_n &= X_n \oplus \Delta_X^{(d)}, \\ \hat{\Gamma}_n^{(d)} &= \Gamma_n^{(d)} \oplus \Delta_\Gamma^{(d)}. \end{aligned} \tag{24}$$

There, $R^{(d)}$ is the number of iterations for each pixel in the diffusion. It is assumed that the encryption starts with the pixel at $(x, y) = (0, 0)$ toward to the last one at $(x, y) = (M-1, N-1)$, so the arrays of bit sequences $\Delta_X^{(d)}$ and $\Delta_\Gamma^{(d)}$ in Figure 4 are

$$\Delta_X^{(d)} = \begin{cases} Y_1^{(d)} \circ C_0 & \text{for } n = 1 \text{ and } (x, y) = (0, 0); \\ Y_1^{(d)} \circ C_{XY} & \text{for } n = 1 \text{ and } (x, y) \neq (0, 0); \\ Y_2^{(d)} \circ X_n & \text{for } 2 \leq n \leq R^{(d)} \text{ and } \forall (x, y), \end{cases} \tag{25}$$

and

$$\Delta_\Gamma^{(d)} = \begin{cases} Y_3^{(d)} \circ C_0 & \text{for } n = 1 \text{ and } (x, y) = (0, 0); \\ Y_3^{(d)} \circ C_{XY} & \text{for } n = 1 \text{ and } (x, y) \neq (0, 0); \\ Y_4^{(d)} \circ X_n & \text{for } 2 \leq n \leq R^{(d)} \text{ and } \forall (x, y), \end{cases} \tag{26}$$

It is noted that C_{XY} is shared between the encryptor and decryptor in the diffusion. After $R^{(d)}$ iterations, the array of bit sequences of ciphered pixels is

$$C_{XY} = (Y_5^{(d)} \circ P_{XY}) \oplus (iY_1^{(d)} \circ X_n). \tag{27}$$

4.4. Inverse Chaotic Diffusion

Similarly, three configurations of Inverse Chaotic Diffusion (iCDs) in the decryptor are illustrated in Figure 5. These are almost identical to those of CDs in Figure 4, except for the additional block Z^{-1} and the ciphertext C_{XY} being interchanged with the plaintext P_{XY} at the output. The block Z^{-1} is to make the cipher data C_{XY} delayed to become C_{XY}^{-1} in the feedback. The equations for $\Delta_X^{(d)}$ and $\Delta_\Gamma^{(d)}$ in the decryptor are

$$\Delta_X^{(d)} = \begin{cases} Y_1^{(d)} \circ C_0 & \text{for } n = 1 \text{ and } (x, y) = (0, 0); \\ Y_1^{(d)} \circ C_{XY}^{-1} & \text{for } n = 1 \text{ and } (x, y) \neq (0, 0); \\ Y_2^{(d)} \circ X_n & \text{for } 2 \leq n \leq R^{(d)} \text{ and } \forall (x, y), \end{cases} \tag{28}$$

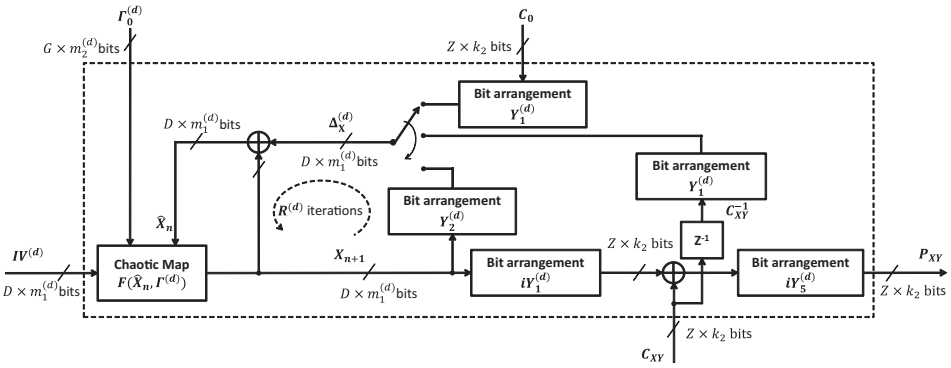
and

$$\Delta_\Gamma^{(d)} = \begin{cases} Y_3^{(d)} \circ C_0 & \text{for } n = 1 \text{ and } (x, y) = (0, 0); \\ Y_3^{(d)} \circ C_{XY}^{-1} & \text{for } n = 1 \text{ and } (x, y) \neq (0, 0); \\ Y_4^{(d)} \circ X_n & \text{for } 2 \leq n \leq R^{(d)} \text{ and } \forall (x, y). \end{cases} \tag{29}$$

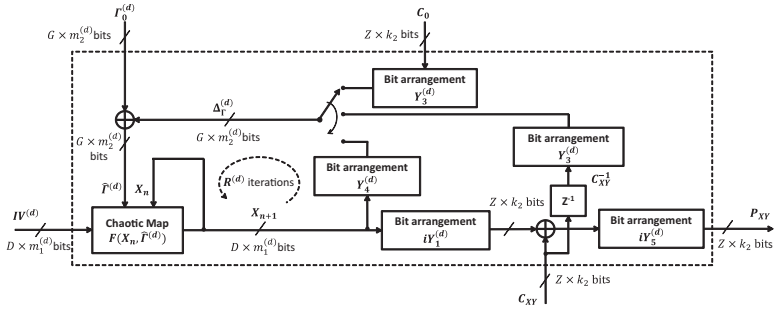
The recovered plain pixels in the form of array of bit sequences after inverse diffusion are

$$P_{XY} = iY_5^{(d)} \circ (C_{XY} \oplus (iY_1^{(d)} \circ X_n)). \tag{30}$$

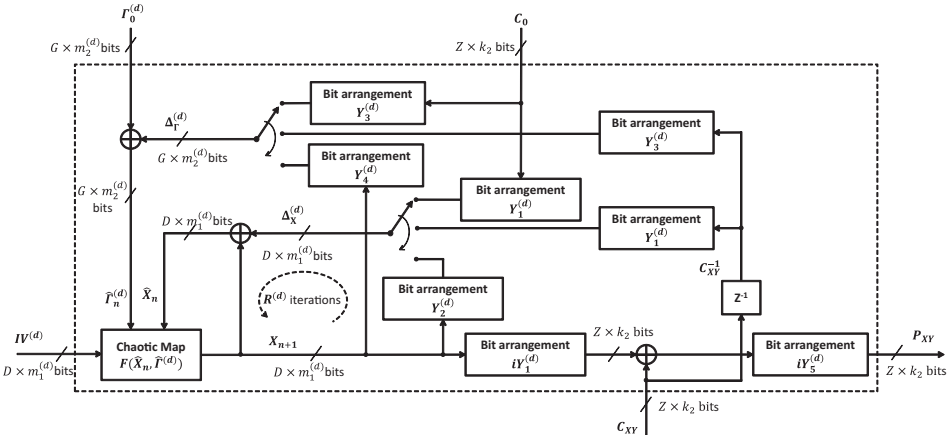
The value of parameters and the operation of iCD are the same as those of CD as described in Section 4.3.



(**a**) Inverse chaotic diffusion with the perturbation on chaotic state variable (iCD-1)



(**b**) Inverse chaotic diffusion with the perturbation on control parameter (iCD-2)



(**c**) Inverse chaotic diffusion with the perturbation on both (iCD-3)

Figure 5. The structure of inverse CD with the perturbation.

4.5. Space of Secret Keys

It is assumed that the number of bits representing for the value of X_n and for that of Γ_n in Figures 3–5 are $D \times m_1^{(d)}$ and $G \times m_2^{(d)}$, respectively. The secret keys of the proposed permutation and diffusion are the value sets of initial vectors of state variables and initial values of control parameters. It is noted that bit arrangements are considered as structural parameters rather than secret keys.

Let us define s_{param} be the number of bits representing for *param*. Table 3 shows the number of bits encoding for values of initial vectors and control parameters for the permutation and diffusion. In fact, the number of bits representing for the secret keys is dependent on the number of perturbed bits in state variables and control parameters. Specifically, the initial value of $IV^{(p)}$, $IV^{(d)}$, C_0, $\Gamma_0^{(p)}$ and $\Gamma_0^{(d)}$) is represented in the format of fixed point and its values are varying in specific ranges. In the scheme

of perturbation on control parameters, the state of some bits in the value of control parameters is fixed to ensure that chaos exhibits while that of the other bits are changeable by perturbation. Similarly in the scheme of perturbation on state variables, some selected bits of state variables are with fixed states while the others are changeable. In other words, a number of bits with fixed states do not contribute to the key space of the permutation and diffusion.

Table 3. The maximum number of bits representing for the initial values.

Parameter	Maximum Number of Bits
$IV^{(p)}$	$s_{IV^{(p)}}$
$IV^{(d)}$	$s_{IV^{(d)}}$
$\Gamma_0^{(p)}$	$s_{\Gamma^{(p)}}$
$\Gamma_0^{(d)}$	$s_{\Gamma^{(d)}}$
C_0	s_{k_2}

However, the number of changeable bits is as large as possible and must be larger than the number of bits encoding for the coordinate and the value of pixels in the appropriate scheme of perturbation.

4.6. Computational Complexity and Resource Analysis

It is emphasized that the cryptosystems working at bit level are designed with the aim to implement on hardware platforms such as Field Programmable Gate Arrays (FPGAs). Here, the computational complexity is considered in the context of using FPGAs, rather than on PC where the basic data unit is byte. In addition, the example will be given to illustrate the computational complexity.

In fact, the computational complexity and resource required for the proposed models are dependent on equations of chaotic maps and a number of bits are used for representing values of state variables and control parameters. The advantage of cryptosystems implemented on the customized hardware is that a number of bits representing for the format of fixed point can be tailored for the requirement of security and application.

The requirement of computational resource is as follows. The chaotic map requires a number of arithmetic operations and logic gates, that is, multipliers, divisors, adders, and subtractors. A number of XOR gates are used for the perturbation. A number of registers are needed to store arithmetic operands and the result. A memory space is necessary to store the plain image, and the permutation and diffusion are performed on this memory. Moreover, as it is implemented on customized hardware, all the blocks of bit arrangement in the proposed models are interconnection wires.

One of advantages in using chaotic maps for image encryption is the low computational complexity. The speed of hardware implementation is dependent mainly on the speed of arithmetic operations of chaotic map, the read/write cycles of memory during the permutation and diffusion.

For example, the Logistic map in Equation (31) is chosen for the scheme of perturbation on state variable. The hardware resource for the Logistic map is as shown in Table 4. Accordingly, it requires four registers, two multipliers, and one substractor. In addition, a number of XOR gates are necessary to implement the perturbation. In fact, it is small necessary resource to implement when it is compared with the available resource of typical FPGA devices.

Table 4. Hardware components to implement the Logistic map.

Term	Register (Buffer)	Multiplier	Substractor
x_n	✓		
a	✓		
$T_1 = a * x_n$	✓	✓	
$T_2 = (1 - x_n)$			✓
$T_1 * T_2$	✓	✓	

5. Example and Simulation

It is noted that this work is to propose the design of permutation and diffusion with different schemes of perturbation, rather than a cryptosystem. In addition, due to the limit of the space, the example is mainly to demonstrate the feasibility of the approach, and only representative samples of simulation results are illustrated in case that other ones are the same. The more detail on the specific application using this approach to design a cryptosystem will be found in other papers published somewhere later.

In this example, the generic 1D Logistic map,

$$x_{n+1} = ax_n(1 - x_n), \tag{31}$$

is employed for both the permutation and diffusion.

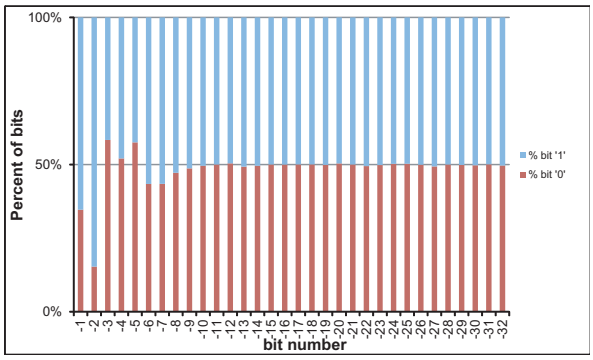
5.1. Percentage of Bits Generated by Logistic Map

The Logistic map in Equation (31) is simulated, in which values of x_n and a are represented in the format of fixed point as 1.32 and 2.32, respectively. Different values of a are as given in Table 5. It is noted that $a = 3.9999$ is written with 4 digits after fraction point, but in fact the exactly value is $4.0 - 2^{-32}$. The initial value of x is 0.1234567890. For each value of a, the Logistic map is iterated 196,608 times to produce chaotic sequences. Due to the value range of $x_n \in (0, 1)$, so the bit b_0 representing for its integer part is always '0'. The percentage of bits (PoB) for bits b_{-1} to b_{-32} and distribution of values (DoV) of chaotic sequences generated by the Logistic map with different values of a are displayed in Figure 6.

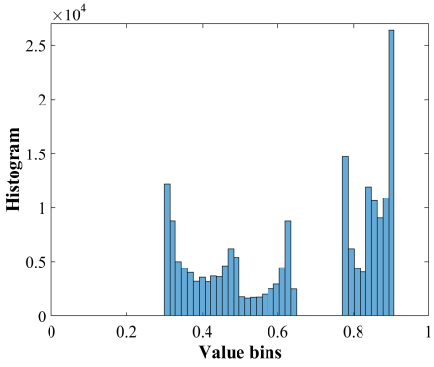
It is clear from Figure 6 that the lower significant bits, that is, from bits b_{-9} to b_{-32}, have PoBs of '0' roughly equal to those of bits '1' for every chosen value of a; whereas the PoBs of b_{-1} to b_{-8} are biased to either '0' or '1'. Besides, DoVs of chaotic sequences are uneven for every value of a. The best DoV is obtained with $a = 3.9999$ as shown in Figure 6q. Intuitively, there is a correlation between PoBs and DoVs. The better DoV is, the better PoBs of higher significant bits are obtained. In addition, PoBs for lower significant bits are independent from DoVs. It suggests that lower significant bits should be utilized for the bit-level encryption, and the value of a should be chosen as close to 4.0 as possible.

Table 5. Chosen values of a for the percentage of bits (PoB) and distribution of values (DoV) analysis.

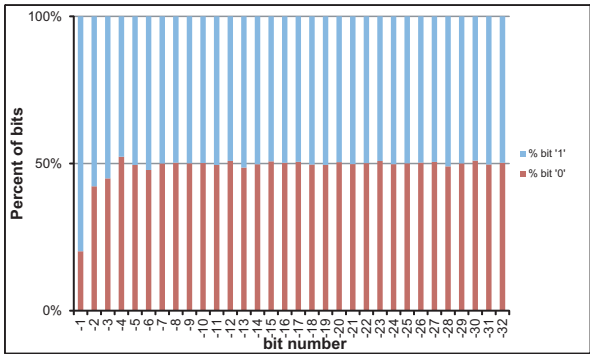
Chosen Values of a	Bit Representation in the Format of 2.32
3.6250	11.10100000000000000000000000000000
3.6875	11.10110000000000000000000000000000
3.7500	11.11000000000000000000000000000000
3.8125	11.11010000000000000000000000000000
3.8750	11.11100000000000000000000000000000
3.9375	11.11110000000000000000000000000000
3.9688	11.11111000000000000000000000000000
3.9844	11.11111100000000000000000000000000
3.9999	11.11111111111111111111111111111111



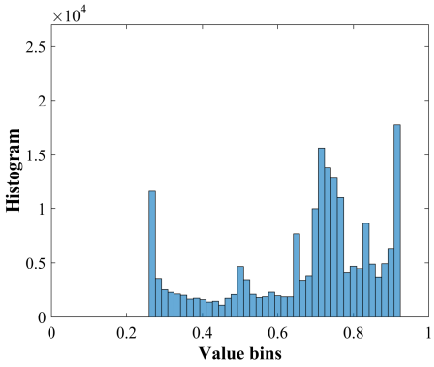
(**a**) PoB with $a = 3.6250$



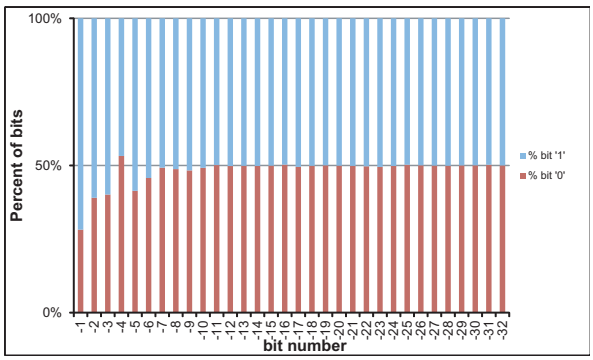
(**b**) DoV with $a = 3.6250$



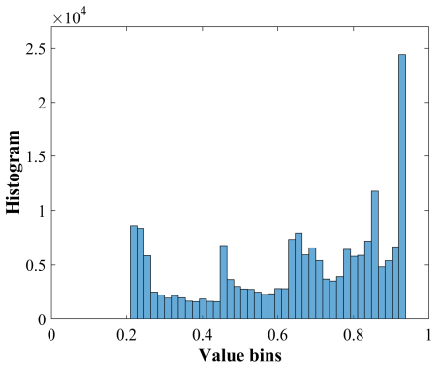
(**c**) PoB with $a = 3.6875$



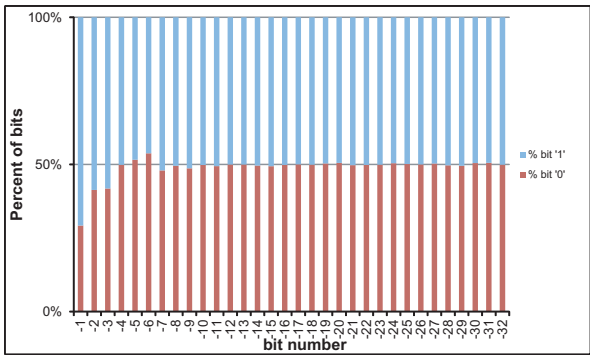
(**d**) DoV with $a = 3.6875$



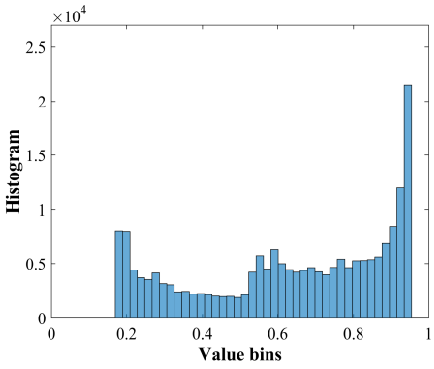
(**e**) PoB with $a = 3.75$



(**f**) DoV with $a = 3.75$

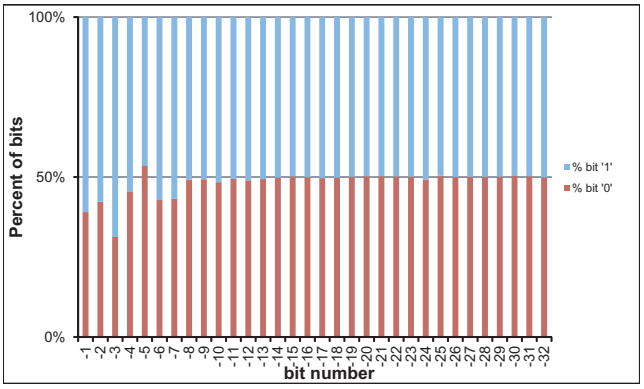


(**g**) PoB with $a = 3.8125$

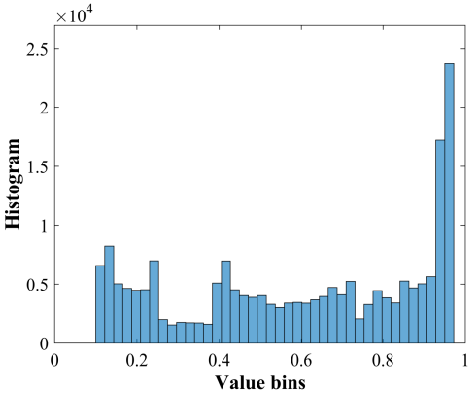


(**h**) DoV with $a = 3.8125$

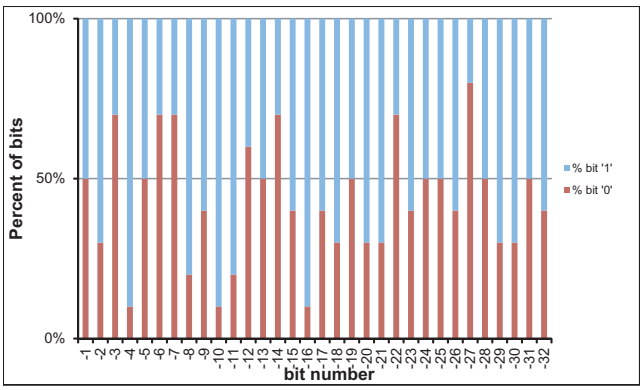
Figure 6. *Cont.*



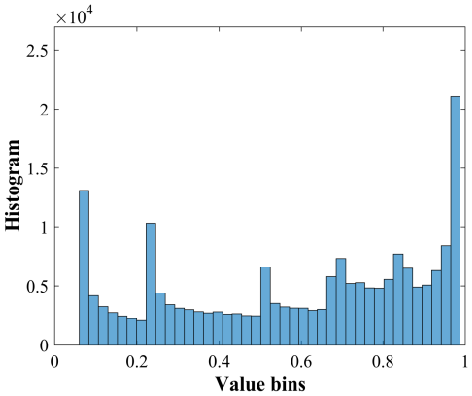
(**i**) PoB with $a = 3.875$



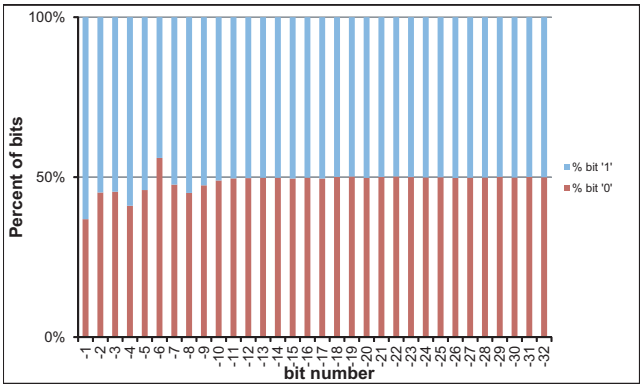
(**j**) DoV with $a = 3.875$



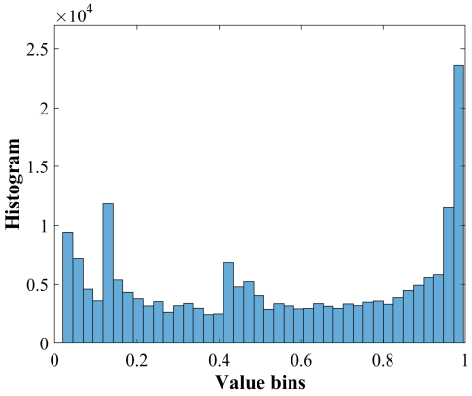
(**k**) PoB with $a = 3.9375$



(**l**) DoV with $a = 3.9375$

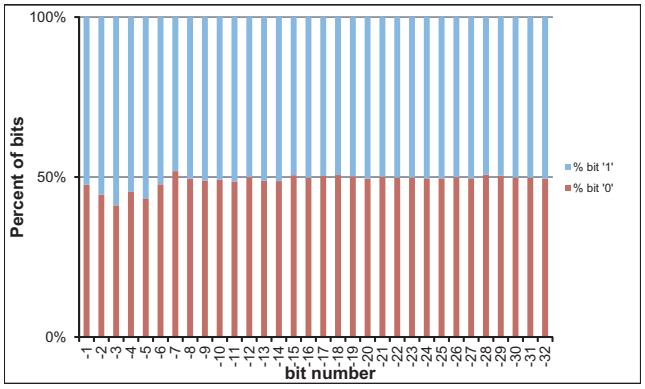


(**m**) PoB with $a = 3.9688$

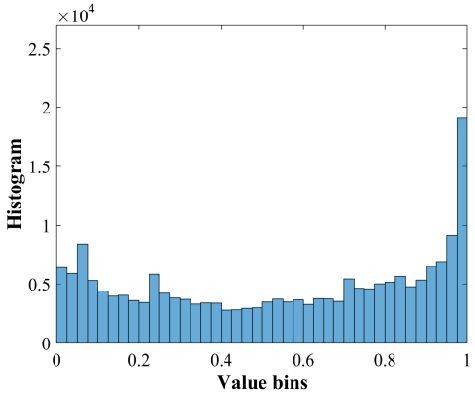


(**n**) DoV with $a = 3.9688$

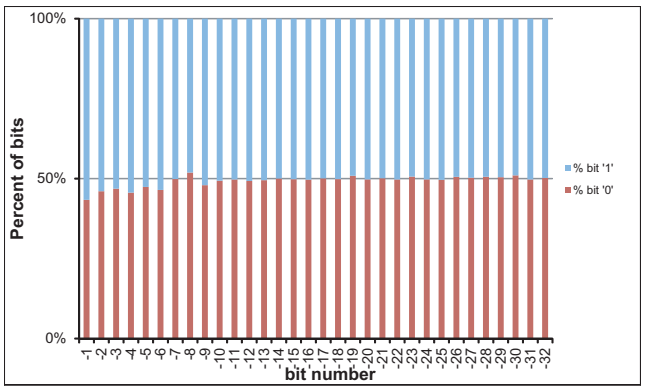
Figure 6. *Cont.*



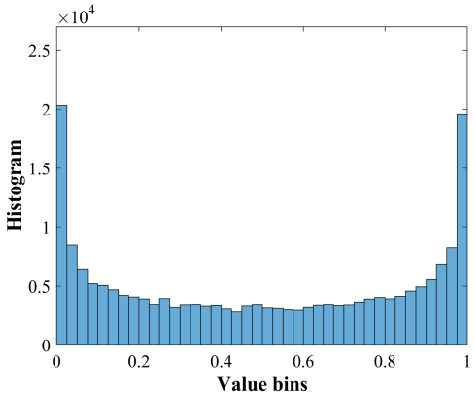
(**o**) PoB with $a = 3.9844$



(**p**) DoV with $a = 3.9844$



(**q**) PoB with $a = 3.9999$



(**r**) DoV with $a = 3.9999$

Figure 6. Percentage of bits and distribution of values of x_n.

5.2. Permutation and Diffusion with Logistic Map

The 1D Logistic map in Equation (31) is employed for the permutation and diffusion, so $D = 1$, and $G = 1$. The notations for state variable and control parameter are $X_n = [x_n]$, $\Gamma_n = [a_n]$, $\Delta_X = [\delta_x]$ and $\Delta_\Gamma = [\delta_a]$. The superscripts (p) and (d) associate with the notations to mention the permutation and diffusion, respectively. It is noted that Logistic map exhibits chaos with $3.56995 \leq a \leq 4.0$, and the value range of x_n is $(0, 1)$.

5.2.1. Chosen Value of Parameters

Values of x_n and a_n in the permutation and diffusion are represented in the format of fixed point as given in Table 6. The format of fixed point for the control parameters $a_n^{(p)}$ and $a_n^{(d)}$, and the state variables $x_n^{(p)}$ and $x_n^{(d)}$ is given in Table 7, in which some bits are with fixed states '0' and '1', and bits denoted by 'x' are perturbed. The bit patterns of state variables and control parameters in Table 7 indicate that the state of bits b_0 of both $x_n^{(p)}$ and $x_n^{(d)}$ is fixed at '0', while that of b_1, b_0, b_{-1} and b_{-3} of $a_n^{(p)}$ and b_1, b_0, b_{-1}, b_{-3} and b_{-4} of $a_n^{(d)}$ is always '1'. The XOR operation is used as the perturbation operator. Therefore, the state of bits in perturbation amounts $\delta_x^{(p)}$, $\delta_x^{(d)}$, $\delta_a^{(p)}$, and $\delta_a^{(d)}$ must be '0' at positions corresponding to bits with fixed states in $x_n^{(p)}$, $x_n^{(d)}$, $a_n^{(p)}$, and $a_n^{(d)}$, respectively.

The initial values of state variables and control parameters are chosen as in Table 8. If the perturbation is applied to, values of state variables and control parameters and amounts of perturbation will vary in the specific ranges as given in Table 9.

Table 6. The number of bits representing for the value of state variables and control parameters of Logistic maps, and for the coordinate and the value of pixels in the permutation and diffusion.

Parameter	No. of Bits	The Format
$m_1^{(p)}$	33	1.32
$m_2^{(p)}$	36	2.34
$m_1^{(d)}$	33	1.32
$m_2^{(d)}$	37	2.35
k_1	16	16.0
k_2	8	8.0

Table 7. Bit patterns of state variables and control parameters.

State Variables & Parameters	Patterns of Bit Representation
$a_n^{(p)}$	11.1×1xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
$a_n^{(d)}$	11.1×11xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
$x_n^{(p)}$	0.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
$x_n^{(d)}$	0.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Table 8. Initial values of cryptosystem's parameters.

Parameter	Initial and Adopted Values
$a_0^{(p)}$	3.6250
$a_0^{(d)}$	3.68750
$IV^{(p)}$	0.0123456789
$IV^{(d)}$	0.9876543210
C_0	123

Table 9. Value ranges of state variables, control parameters, and amounts of perturbation.

State Variables, Control Parameters and Amounts of Perturbation	Value Ranges
$\hat{a}_n^{(p)}$	$[3.6250, 3.7500)$ and $[3.8750, 4.0)$
$\delta_a^{(p)}$	$[0, 0.1250)$ and $[0.2500, 0.3750)$
$\hat{a}_n^{(d)}$	$[3.6875, 3.7500)$ and $[3.9375, 4.0)$
$\delta_a^{(d)}$	$[0, 0.0625)$ and $[0.2500, 0.3125)$
$\hat{x}_n^{(p)}$	$(0, 1)$
$\delta_x^{(p)}$	$(0, 1)$
$\hat{x}_n^{(d)}$	$(0, 1)$
$\delta_x^{(d)}$	$(0, 1)$

For simplest assumption, let us represent the coordinate and the value of pixels by a 1D sequence of bits, or $Q = 1$ and $Z = 1$. The 8-bit grayscale images with the size of 256×256 are encrypted, so the row and column numbers are encoded by 8 bits. In other words, $XY_{present}$ is represented by a bit sequence of $k_1 = 16$ bits as $(b_{15}b_{14}b_{13}b_{12}b_{11}b_{10}b_9b_8b_7b_6b_5b_4b_3b_2b_1b_0)$ in which the sequences $(b_{15}...b_8)$ and $(b_7...b_0)$ are encoded for values of $x_{present}$ and $y_{present}$, respectively. The value of pixels is represented by a sequence of 8 bits, that is, $Z = 1$ and $k_2 = 8$. Therefore, The bit arrangements are chosen as in Table 10, in which the bits with fixed states in the state variables and control parameters have the position indicated by BIT_0. It is noted from the bit arrangements $Y_2^{(p)}$, $Y_4^{(p)}$, $Y_2^{(d)}$ and $Y_4^{(d)}$ in Table 10 that the bits with poor PoBs in x_n as displayed in Figure 6 are deliberately used in the permutation and diffusion to consolidate the suggestion to utilize lower significant bits in the encryption.

Table 10. Matrices of bit arrangement.

$Y_1^{(p)}$	[BIT_0 (1.1) (1.9) (1.2) (1.10) (1.3) (1.11) (1.4) (1.12) (1.5) (1.13) (1.6) (1.14) (1.7) (1.15) (1.8) (1.16) (1.1) (1.9) (1.2) (1.10) (1.3) (1.11) (1.4) (1.12) (1.5) (1.13) (1.6) (1.14) (1.7) (1.15) (1.8) (1.16)]
$iY_1^{(p)}$	[(1.30) (1.18) (1.12) (1.20) (1.16) (1.25) (1.19) (1.13) (1.24) (1.11) (1.26) (1.17) (1.21) (1.24) (1.22) (1.27)]
$Y_2^{(p)}$	[BIT_0 (1.33) (1.32) (1.31) (1.30) (1.29) (1.28) (1.27) (1.26) (1.25) (1.24) (1.23) (1.22) (1.21) (1.20) (1.19) (1.18) (1.17) (1.16) (1.15) (1.14) (1.13) (1.12) (1.11) (1.10) (1.9) (1.8) (1.7) (1.6) (1.5) (1.4) (1.3) (1.2)]
$Y_3^{(p)}$	[BIT_0 BIT_0 BIT_0 (1.8) BIT_0 (1.16) (1.7) (1.15) (1.6) (1.14) (1.5) (1.13) (1.4) (1.12) (1.3) (1.11) (1.2) (1.10) (1.1) (1.9) (1.8) (1.16) (1.7) (1.15) (1.6) (1.14) (1.5) (1.13) (1.4) (1.12) (1.3) (1.11) (1.2) (1.10) (1.1) (1.9)]
$Y_4^{(p)}$	[BIT_0 BIT_0 BIT_0 (1.33) BIT_0 (1.32) (1.31) (1.30) (1.29) (1.28) (1.27) (1.26) (1.25) (1.24) (1.23) (1.22) (1.21) (1.20) (1.19) (1.18) (1.17) (1.16) (1.15) (1.14) (1.13) (1.12) (1.11) (1.10) (1.9) (1.8) (1.7) (1.6) (1.5) (1.4) (1.3) (1.2)]
$Y_1^{(d)}$	[BIT_0 (1.8) (1.2) (1.5) (1.1) (1.7) (1.2) (1.6) (1.4) (1.5) (1.3) (1.7) (1.8) (1.4) (1.2) (1.7) (1.1) (1.5) (1.3) (1.6) (1.2) (1.4) (1.8) (1.1) (1.3) (1.4) (1.6) (1.5) (1.8) (1.6) (1.1) (1.7) (1.3)]
$iY_1^{(d)}$	[(1.3) (1.8) (1.2) (1.20) (1.16) (1.6) (1.21) (1.30)]
$Y_2^{(d)}$	[BIT_0 (1.16) (1.11) (1.29) (1.32) (1.18) (1.13) (1.10) (1.7) (1.14) (1.31) (1.4) (1.12) (1.26) (1.5) (1.17) (1.9) (1.22) (1.24) (1.15) (1.21) (1.28) (1.23) (1.6) (1.33) (1.19) (1.8) (1.30) (1.2) (1.3) (1.27) (1.20) (1.25)]
$Y_3^{(d)}$	[BIT_0 BIT_0 BIT_0 (1.8) BIT_0 BIT_0 (1.7) (1.6) (1.5) (1.4) (1.3) (1.2) (1.1) (1.1) (1.2) (1.3) (1.4) (1.5) (1.6) (1.7) (1.8) (1.1) (1.2) (1.3) (1.4) (1.5) (1.6) (1.7) (1.8) (1.1) (1.2) (1.3) (1.4) (1.5) (1.6) (1.7) (1.8)]
$Y_4^{(d)}$	[BIT_0 BIT_0 BIT_0 (1.33) BIT_0 BIT_0 (1.32) (1.31) (1.30) (1.29) (1.28) (1.27) (1.26) (1.25) (1.24) (1.23) (1.22) (1.21) (1.20) (1.19) (1.18) (1.2) (1.3) (1.4) (1.5) (1.6) (1.7) (1.8) (1.9) (1.10) (1.11) (1.12) (1.13) (1.14) (1.15) (1.16) (1.17)]
$Y_5^{(d)}$	[(1.8) (1.7) (1.6) (1.5) (1.1) (1.2) (1.3) (1.4)]
$iY_5^{(d)}$	[(1.5) (1.6) (1.7) (1.8) (1.4) (1.3) (1.2) (1.1)]

In this example, the four 8-bit grayscale images [69] and two special ones with the size of 256×256 are used for the simulation, that is, Lena, Cameraman, House, and Peppers, Black and White. The simulation is carried out for the permutation and diffusion separately, and the input of the permutation and diffusion processes are the original images. The value of other parameters is chosen as: the number of iterations for each data unit in the permutation and diffusion is $R^{(p)} = 10$ and $R^{(d)} = 10$, respectively; and the number of permutation and diffusion rounds is $N^{(p)} = 3$ and $N^{(d)} = 3$.

Next, the simulation results is to show to effectiveness of the proposed schemes by means of the PoBs and DoVs of perturbed state variables and control parameters.

5.2.2. Simulation Result of Permutation with Perturbation

The PoB and DoV are measured for values of state variables, control parameters as well as amounts of perturbation in the permutation and diffusion processes. It is noted that only significant samples of results are illustrated representatively to save the space.

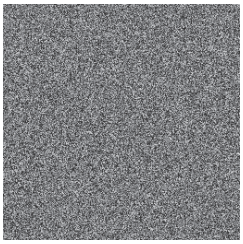
Permuted images with the perturbation on the state variable, control parameter, and on both are illustrated in Figures 7–9. The first column displays the original images, and the second, third and fourth columns are permuted images with different number of permutation rounds $N^{(p)} = 1$, 2, and 3, respectively. It is clear that the visual structure of the original images are completely removed in the permuted images, even after the first round of permutation.

Let us analyze the DoV for state variable and control parameter, and amounts of perturbation for each scheme of perturbation. Specifically, the analysis is carried out with the chaotic sequence $\hat{x}_n^{(p)}$ and the amount of perturbation $\delta_x^{(p)}$ for the perturbation on state variable; with the value sequence of control parameter of $\hat{a}_n^{(p)}$ and the amount of perturbation of $\delta_a^{(p)}$ for the perturbation on control parameter; and with the chaotic sequence of $\hat{x}_n^{(p)}$, the value sequence of control parameter of $\hat{a}_n^{(p)}$, the amounts of perturbation of $\delta_x^{(p)}$ and $\delta_a^{(p)}$ for the perturbation on both. The PoBs for the perturbation amounts are also shown in all schemes of perturbation.

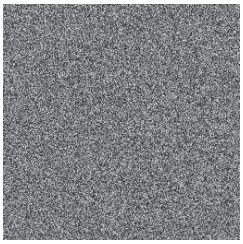
Table 7 shows the chosen pattern of bit representation for $a_n^{(p)}$ to explain the bias of bits in PoBs. There are some bits with the fixed state of '1' and some perturbed bits with 'x'. Due to the fixed state of bits, the value range of control parameter is broken apart to separate portions as given in Table 9, and it can be seen in Figures 11b,c and 12e,f in presenting the DoVs of control parameters and amounts of perturbation.

Figures 10–12 illustrate for the PoBs and DoVs in the perturbation on state variable, control parameter, and on both, respectively. The PoBs are displayed in the first column, and the DoVs are in the second and third columns. Notably, the permutation process uses the coordinates of pixels (x, y) as the input. In addition, for any image with the same size, the permutation rule is the same in every permutation round for any image with the same size, or it is independent from the pixel values. Thus, for each of images, the PoBs and DoVs of the first round of permutation are shown.

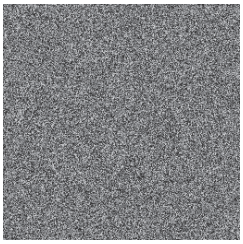
(**a**) Original Lena



(**b**) Lena with $N^{(p)} = 1$

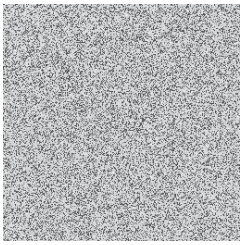


(**c**) Lena with $N^{(p)} = 2$

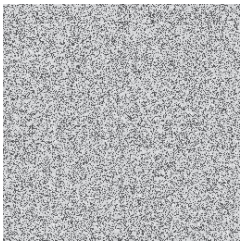


(**d**) Lena with $N^{(p)} = 3$

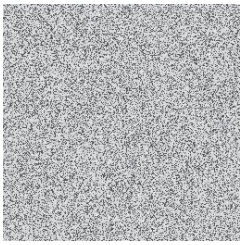
(**e**) Original Cameraman



(**f**) Cameraman with $N^{(p)} = 1$

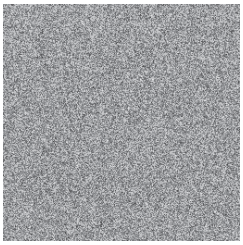


(**g**) Cameraman with $N^{(p)} = 2$



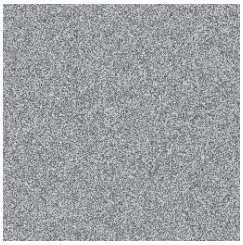
(**h**) Cameraman with $N^{(p)} = 3$

(**i**) Original House



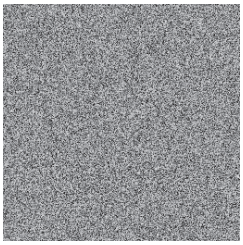
(**j**) House with $N^{(p)} = 1$

(**k**) House with $N^{(p)} = 2$

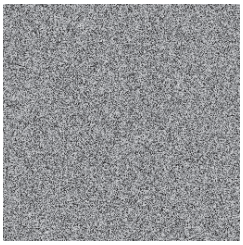


(**l**) House with $N^{(p)} = 3$

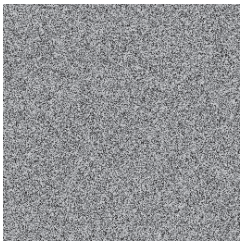
(**m**) Original Peppers



(**n**) Peppers with $N^{(p)} = 1$



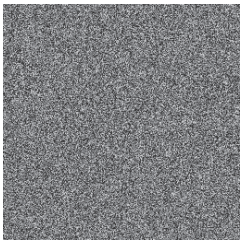
(**o**) Peppers with $N^{(p)} = 2$



(**p**) Peppers with $N^{(p)} = 3$

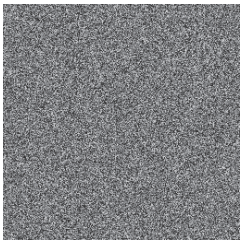
Figure 7. The permuted images with the perturbation on state variable.

(**a**) Original Lena



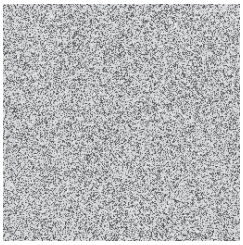
(**b**) Lena with $N^{(p)} = 1$

(**c**) Lena with $N^{(p)} = 2$

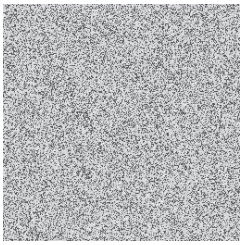


(**d**) Lena with $N^{(p)} = 3$

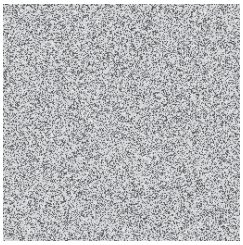
(**e**) Original Cameraman



(**f**) Cameraman with $N^{(p)} = 1$

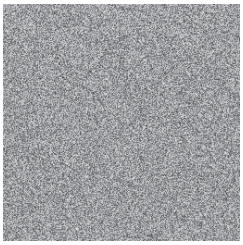


(**g**) Cameraman with $N^{(p)} = 2$

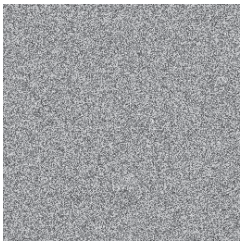


(**h**) Cameraman with $N^{(p)} = 3$

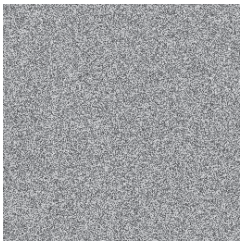
(**i**) Original House



(**j**) House with $N^{(p)} = 1$



(**k**) House with $N^{(p)} = 2$

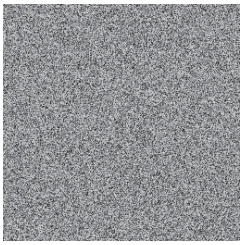


(**l**) House with $N^{(p)} = 3$

(**m**) Original Peppers

(**n**) Peppers with $N^{(p)} = 1$

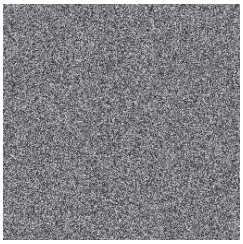
(**o**) Peppers with $N^{(p)} = 2$



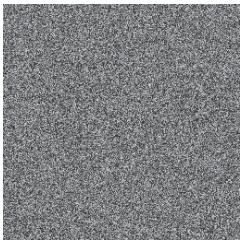
(**p**) Peppers with $N^{(p)} = 3$

Figure 8. The permuted images with the perturbation on control parameter.

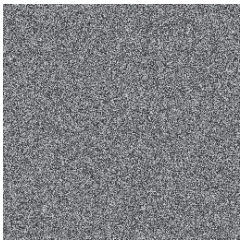
(**a**) Original Lena



(**b**) Lena with $N^{(p)} = 1$

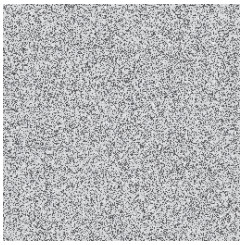


(**c**) Lena with $N^{(p)} = 2$

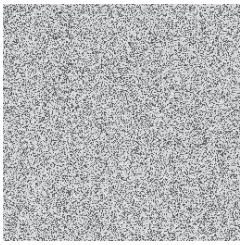


(**d**) Lena with $N^{(p)} = 3$

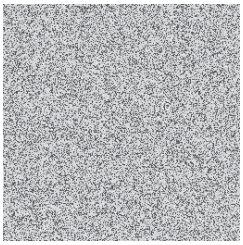
(**e**) Original Cameraman



(**f**) Cameraman with $N^{(p)} = 1$

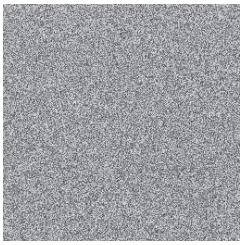


(**g**) Cameraman with $N^{(p)} = 2$

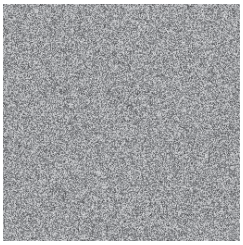


(**h**) Cameraman with $N^{(p)} = 3$

(**i**) Original House



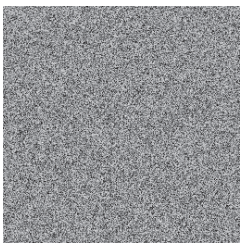
(**j**) House with $N^{(p)} = 1$



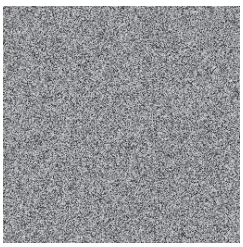
(**k**) House with $N^{(p)} = 2$

(**l**) House with $N^{(p)} = 3$

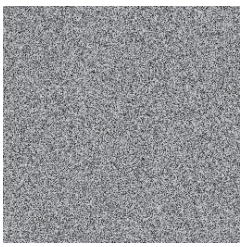
(**m**) Original Peppers



(**n**) Peppers with $N^{(p)} = 1$

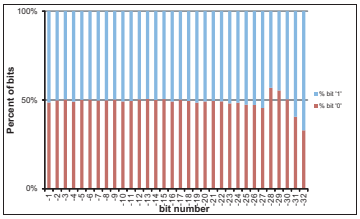


(**o**) Peppers with $N^{(p)} = 2$

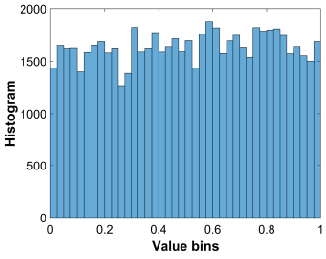


(**p**) Peppers with $N^{(p)} = 3$

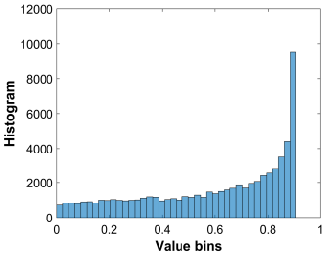
Figure 9. The permuted images with the perturbation on both.



(**a**) PoB in $\delta_x^{(p)}$ at round 1

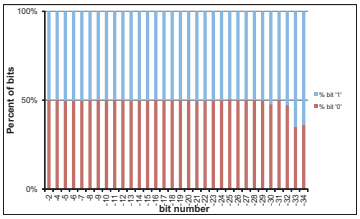


(**b**) DoV of $\delta_x^{(p)}$ at round 1

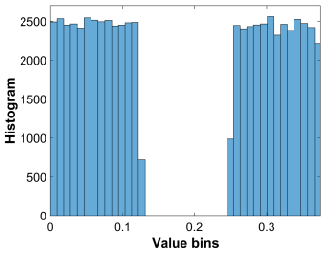


(**c**) DoV of $\hat{x}_n^{(p)}$ at round 1

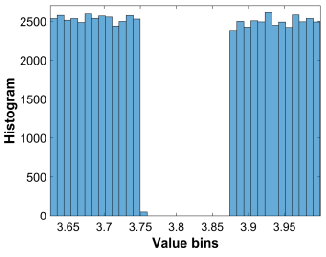
Figure 10. Permutation with the perturbation on state variable: PoB and DoV of amount of permutation and perturbed state variable.



(**a**) PoB in $\delta_a^{(p)}$ at round 1

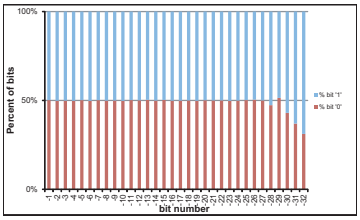


(**b**) DoV of $\delta_a^{(p)}$ at round 1

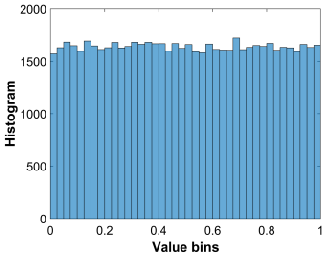


(**c**) DoV of $\hat{a}_n^{(p)}$ at round 1

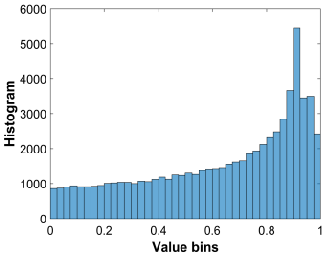
Figure 11. Permutation with the perturbation on control parameter: PoB and DoV of amount of permutation and perturbed control parameter.



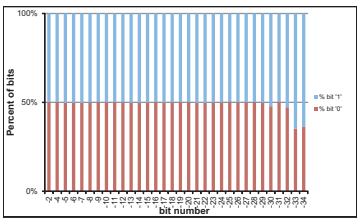
(**a**) PoB in $\delta_x^{(p)}$ at round 1



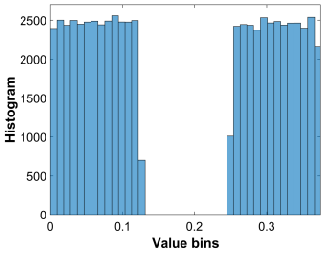
(**b**) DoV of $\delta_x^{(p)}$ at round 1



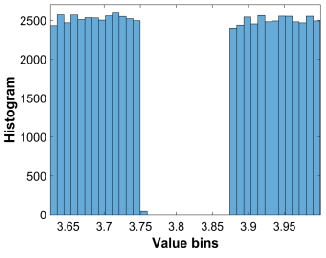
(**c**) DoV of $\hat{x}_n^{(p)}$ at round 1



(**d**) PoB in $\delta_a^{(p)}$ at round 1



(**e**) DoV of $\delta_a^{(p)}$ at round 1



(**f**) DoV of $\hat{a}_n^{(p)}$ at round 1

Figure 12. Permutation with the perturbation on both: PoB and DoV of amounts of permutation, and perturbed state variable and control parameter.

It is clear from the first column of Figures 10–12 that the PoBs of amounts of perturbation are even for most significant bits, and it is biased for a few lower significant bits in every scheme of perturbation. That is because the higher significant bits of $x_n^{(p)}$ are utilized to construct the amounts of perturbation $\delta_x^{(p)}$ and $\delta_a^{(p)}$ by the bit arrangement rules $Y_2^{(p)}$ and $Y_4^{(p)}$ in Table 10. This is agreed with the PoBs of x_n as shown in Figure 6. In other words, the lower significant bits of $x_n^{(p)}$ should be employed to generate amounts of perturbation.

The DoVs of amounts of perturbation $\delta_x^{(p)}$ are spread over the range of (0,1) for the perturbation on state variable and on both as depicted in Figures 10b and 12b. In contrast, the DoVs of perturbed state variable $\hat{x}_n^{(p)}$ cover the lower range of (0,1) for the perturbation on state variable in Figure 10c and the full range of (0,1) for the perturbation on both in Figure 12c. In addition, the DoVs of $\delta_x^{(p)}$ in the schemes of perturbation on state variable and on both are fairly flat while that of $\hat{x}_n^{(p)}$ is not.

As demonstrated in Figures 11 and 12, the DoVs of control parameter $\hat{a}_n^{(p)}$ and its amounts of perturbation $\delta_x^{(p)}$ and $\delta_a^{(p)}$ do not cover full range of (0,1) because the bit pattern of $a_n^{(p)}$ is chosen as in Table 7. The bits at the positions b_1, b_0, b_{-1}, b_{-3} are fixed at the state '1', while bits at b_{-2}, b_{-4},...b_{-34} are perturbed. This makes the value range of control parameter reduced and partitioned apart. One perturbed bit in-between of two fixed bits, that is, the bit b_{-2}, in the fractional portion of the bit pattern of $a_n^{(p)}$ makes the value ranges of $\delta_a^{(p)}$ and $\hat{a}_n^{(p)}$ divided into two separate portions as shown Figures 11 and 12. That is agreed with the portions of value ranges given in Table 9. In general, there are 2^{n_b} separate portions of value ranges for n_b perturbed bits in-between fixed bits.

5.2.3. Simulation Result of Diffusion with Perturbation

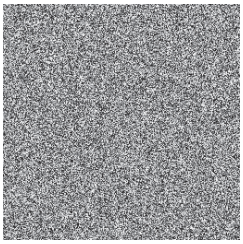
Figures 13–15 illustrate the original images and its corresponding diffused ones in the second, third, and fourth columns with different number of diffusion rounds, that is, $N^{(d)} = 1$, 2, and 3. Note that each pixel is iterated ten times ($R^{(d)} = 10$). It is clear that the visual structure of the original images is completely destroyed in the diffused images, even after the first round of diffusion.

To save the space, the PoBs and DoVs in the diffusion of only Cameraman image are illustrated in Figures 16–19. The result shows almost the same to those in the permutation as described above.

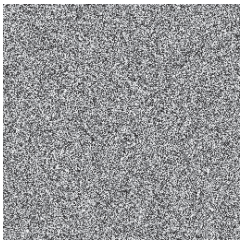
The PoBs in the first column shows the bias to bit '1' at the bit positions b_{-28} and b_{-29} of $\delta_x^{(d)}$ in Figures 16 and 18, and at b_{-20}, b_{-21}, b_{-22}, b_{-24} and b_{-25} of $\delta_a^{(d)}$ in Figures 17 and 19. The bias also occurs to bit '0' at the bit positions b_{-8} and b_{-14} of $\delta_x^{(d)}$ in Figure 16. As described by $Y_2^{(d)}$ and $Y_4^{(d)}$ in Table 10, the bias is caused by higher significant bits of $\hat{x}_n^{(d)}$ employed to construct the amounts of perturbation $\delta_x^{(d)}$ and $\delta_a^{(d)}$. This is also agreed with the PoBs of x_n as shown in Figure 6. Similar to above permutation, the lower significant bits of $x_n^{(d)}$ should be chosen to generate amounts of perturbation in the diffusion.

(**a**) Original Lena

(**b**) Lena at round 1

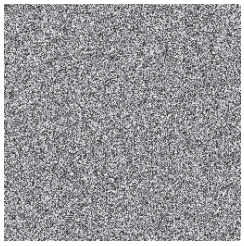


(**c**) Lena at round 2

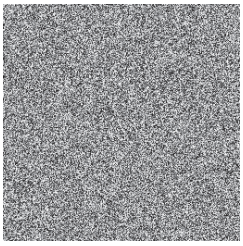


(**d**) Lena at round 3

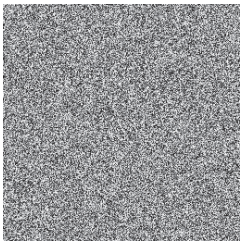
(**e**) Original Cameraman



(**f**) Cameraman at round 1

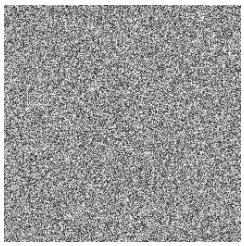


(**g**) Cameraman at round 2

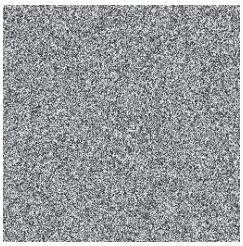


(**h**) Cameraman at round 3

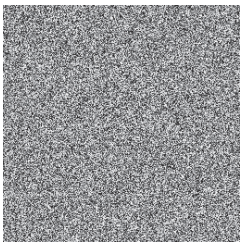
(**i**) Original House



(**j**) House at round 1

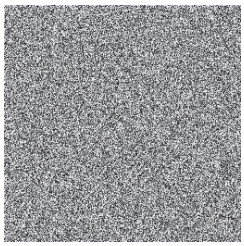


(**k**) House at round 2

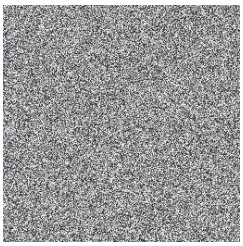


(**l**) House at round 3

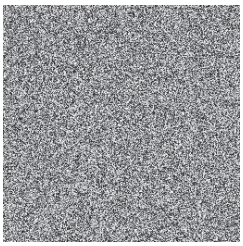
(**m**) Original Peppers



(**n**) Peppers at round 1

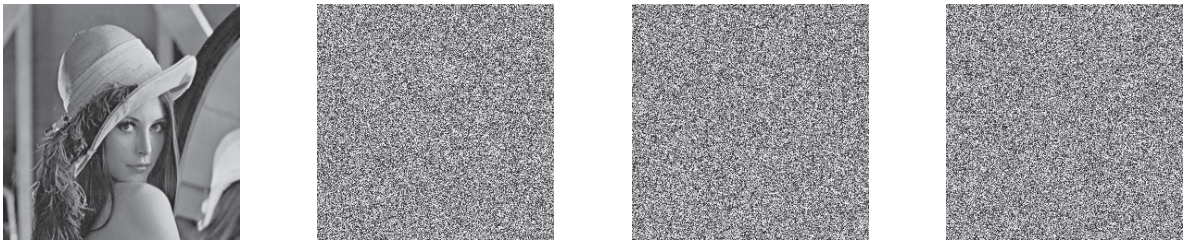


(**o**) Peppers at round 2



(**p**) Peppers at round 3

Figure 13. The diffused images with the perturbation on state variable.



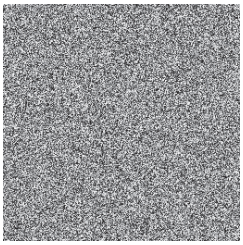
(**a**) Original Lena

(**b**) Lena at round 1

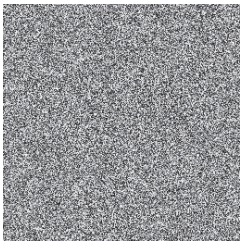
(**c**) Lena at round 2

(**d**) Lena at round 3

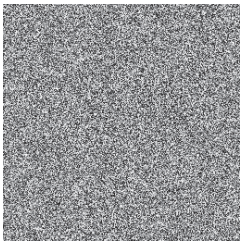
(**e**) Original Cameraman



(**f**) Cameraman at round 1

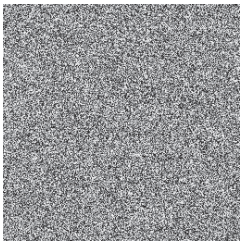


(**g**) Cameraman at round 2

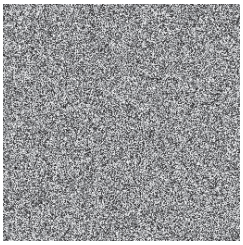


(**h**) Cameraman at round 3

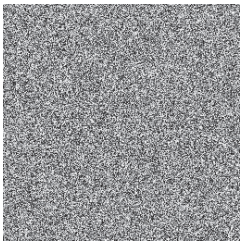
(**i**) Original House



(**j**) House at round 1

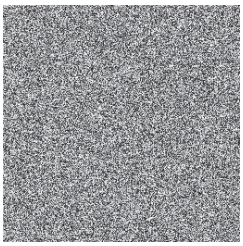


(**k**) House at round 2

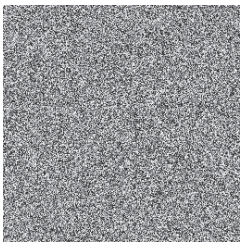


(**l**) House at round 3

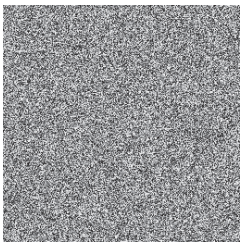
(**m**) Original Peppers



(**n**) Peppers at round 1



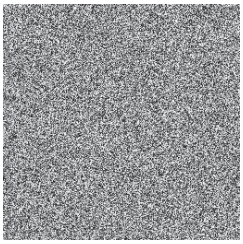
(**o**) Peppers at round 2



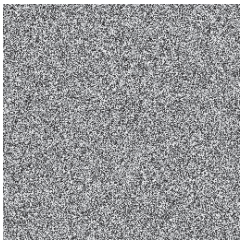
(**p**) Peppers at round 3

Figure 14. The diffused images with the perturbation on control parameter.

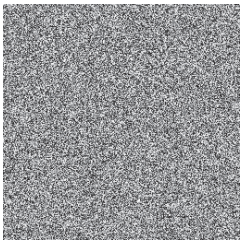
(**a**) Original Lena



(**b**) Lena at round 1

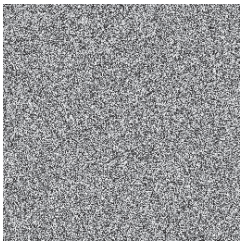


(**c**) Lena at round 2

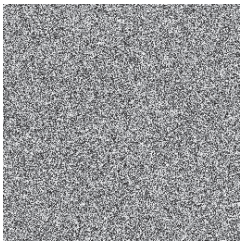


(**d**) Lena at round 3

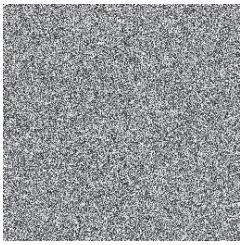
(**e**) Original Cameraman



(**f**) Cameraman at round 1

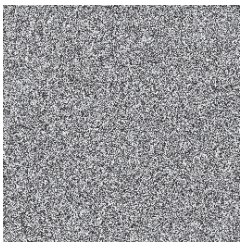


(**g**) Cameraman at round 2

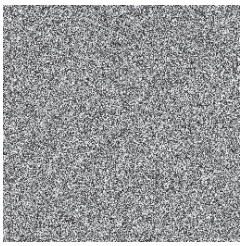


(**h**) Cameraman at round 3

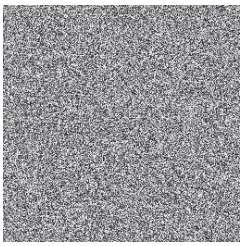
(**i**) Original House



(**j**) House at round 1

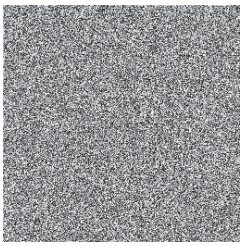


(**k**) House at round 2

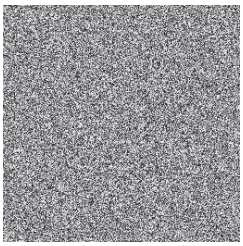


(**l**) House at round 3

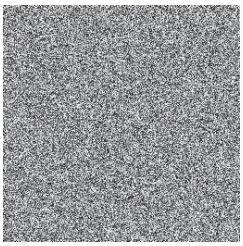
(**m**) Original Peppers



(**n**) Peppers at round 1

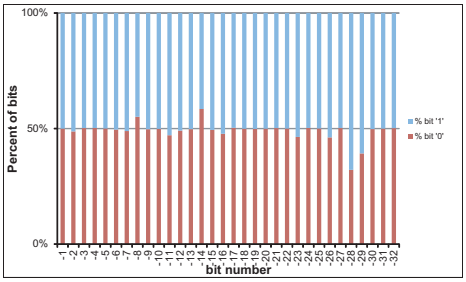


(**o**) Peppers at round 2

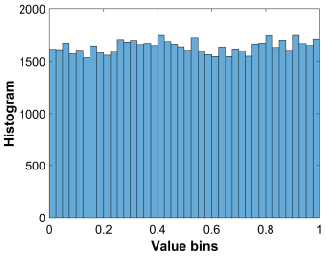


(**p**) Peppers at round 3

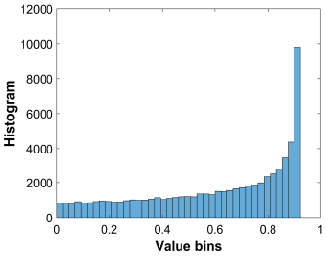
Figure 15. The diffused images with the perturbation on both.



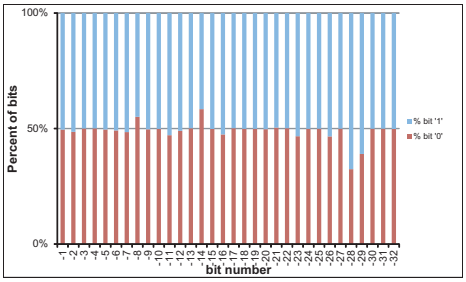
(**a**) Percentage of bits in $\delta_x^{(d)}$ at round 1



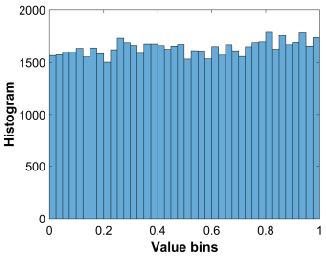
(**b**) DoV of $\delta_x^{(d)}$ at round 1



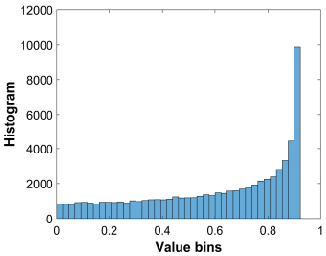
(**c**) DoV of $\hat{x}^{(d)}$ at round 1



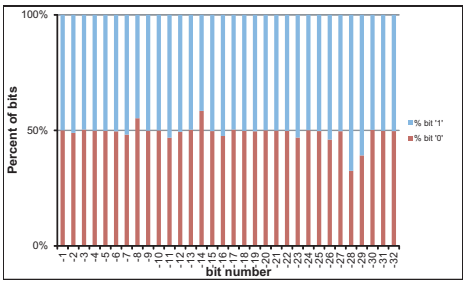
(**d**) PoB of $\delta_x^{(d)}$ at round 2



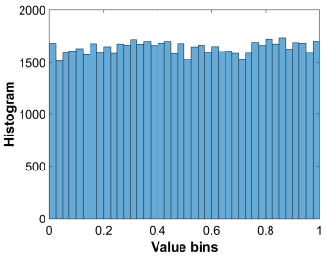
(**e**) DoV of $\delta_x^{(d)}$ at round 2



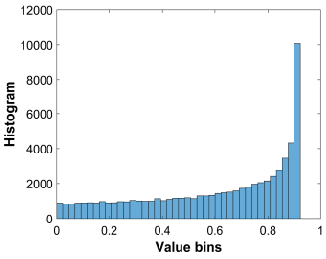
(**f**) DoV of $\hat{x}^{(d)}$ at round 2



(**g**) PoB of $\delta_x^{(d)}$ at round 3

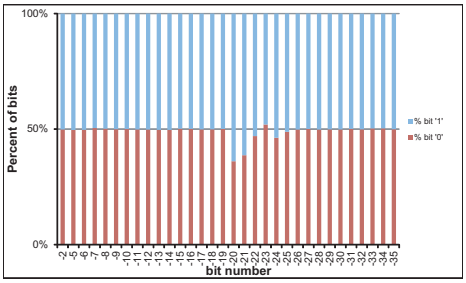


(**h**) DoV of $\delta_x^{(d)}$ at round 3

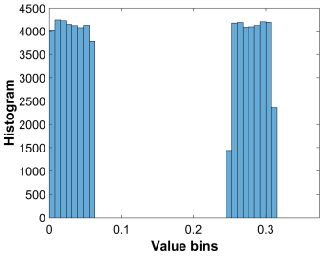


(**i**) DoV of $\hat{x}^{(d)}$ at round 3

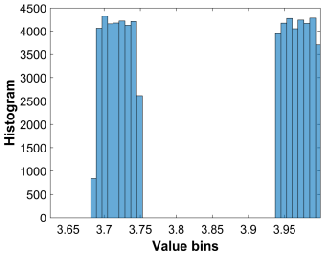
Figure 16. Diffused Cameraman: PoBs and DoVs with the perturbation on state variable.



(**a**) Percentage of bits in $\delta_a^{(d)}$ at round 1

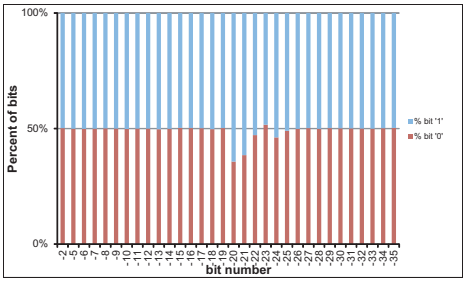


(**b**) DoV of $\delta_a^{(d)}$ at round 1

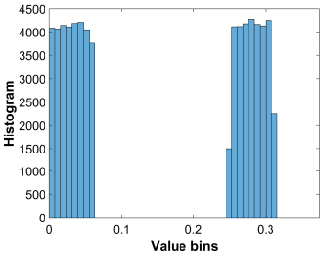


(**c**) DoV of $\hat{a}^{(d)}$ at round 1

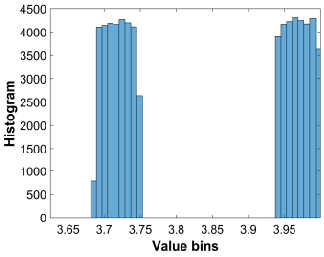
Figure 17. *Cont.*



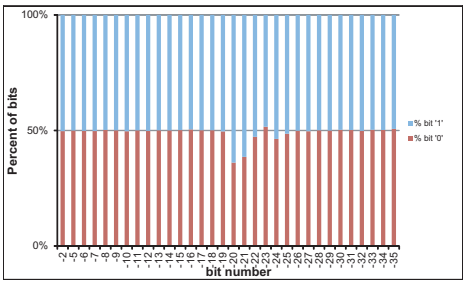
(**d**) PoB of $\delta_a^{(d)}$ at round 2



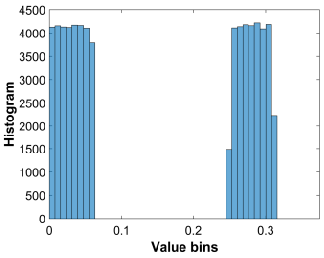
(**e**) DoV of $\delta_a^{(d)}$ at round 2



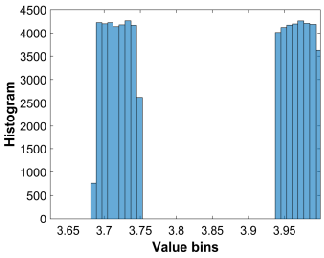
(**f**) DoV of $\hat{a}^{(d)}$ at round 2



(**g**) PoB of $\delta_a^{(d)}$ at round 3

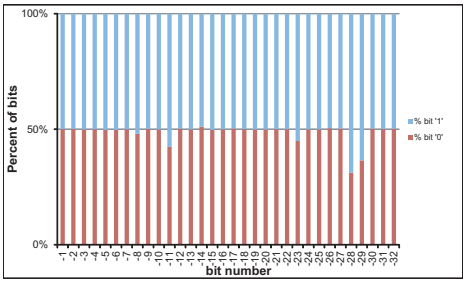


(**h**) DoV of $\delta_a^{(d)}$ at round 3

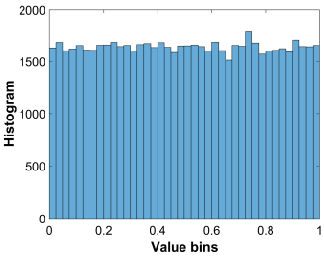


(**i**) DoV of $\hat{a}^{(d)}$ at round 3

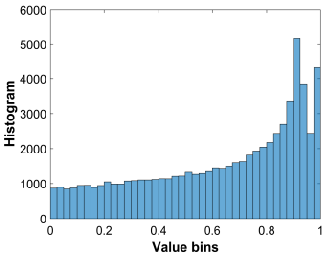
Figure 17. Diffused Cameraman: PoBs and DoVs with the perturbation on control parameter.



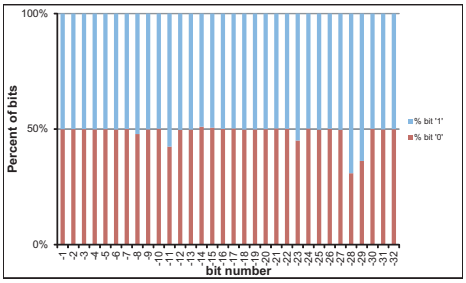
(**a**) Percentage of bits in $\delta_x^{(d)}$ at round 1



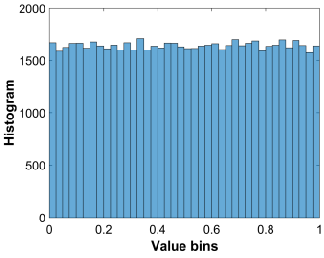
(**b**) DoV of $\delta_x^{(d)}$ at round 1



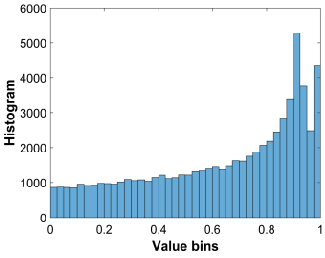
(**c**) DoV of $\hat{x}^{(d)}$ at round 1



(**d**) PoB of $\delta_x^{(d)}$ at round 2

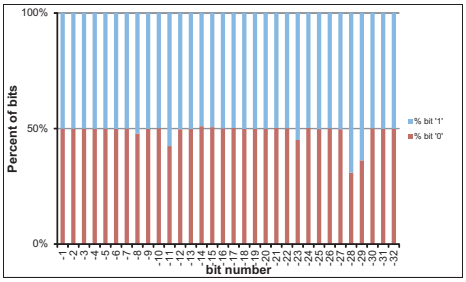


(**e**) DoV of $\delta_x^{(d)}$ at round 2

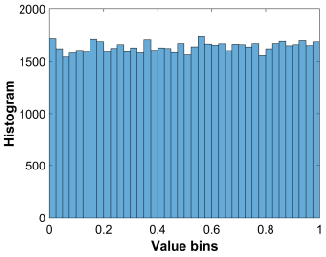


(**f**) DoV of $\hat{x}^{(d)}$ at round 2

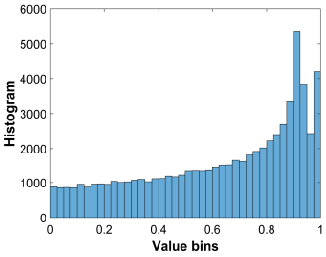
Figure 18. *Cont.*



(**g**) PoB of $\delta_x^{(d)}$ at round 3

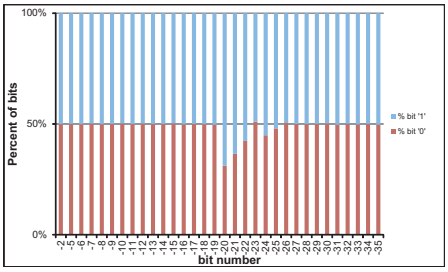


(**h**) DoV of $\delta_x^{(d)}$ at round 3

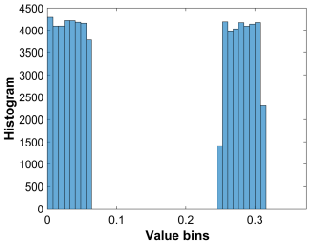


(**i**) DoV of $\hat{x}^{(d)}$ at round 3

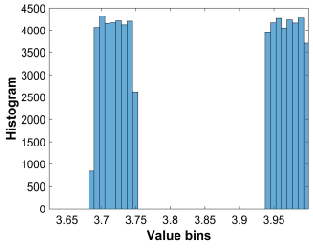
Figure 18. Diffused Cameraman: PoBs and DoVs of state variable with the perturbation on both.



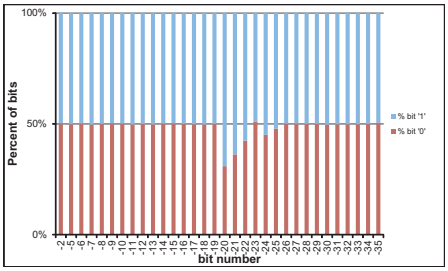
(**a**) Percentage of bits in $\delta_a^{(d)}$ at round 1



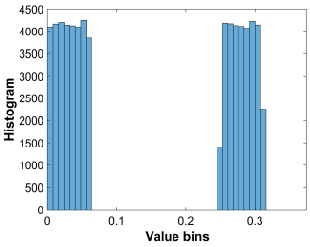
(**b**) DoV of $\delta_a^{(d)}$ at round 1



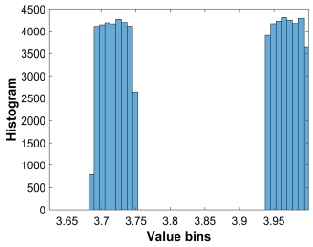
(**c**) DoV of $\hat{a}^{(d)}$ at round 1



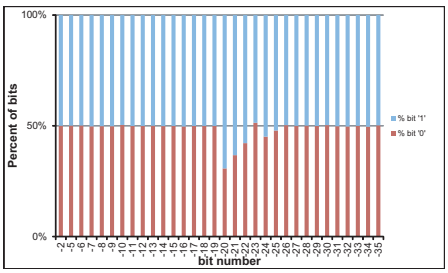
(**d**) PoB of $\delta_a^{(d)}$ at round 2



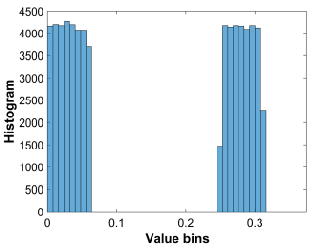
(**e**) DoV of $\delta_a^{(d)}$ at round 2



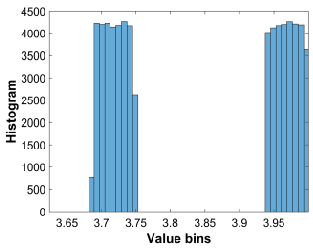
(**f**) DoV of $\hat{a}^{(d)}$ at round 2



(**g**) PoB of $\delta_a^{(d)}$ at round 3



(**h**) DoV of $\delta_a^{(d)}$ at round 3



(**i**) DoV of $\hat{a}^{(d)}$ at round 3

Figure 19. Diffused Cameraman: PoBs and DoVs of control parameter with the perturbation on both.

5.2.4. Space of Secret Keys

The secret keys in the proposed permutation and diffusion are the initial values of state variables and control parameters. In fact, values of state variables and control parameters are changed during perturbation. The Logistic map in chaotic behavior requires the control parameter and the state variable varying in defined ranges. That is, the integer portions of values of state variables and control

parameters must be '0' and '11', respectively. In addition, there are some bits in the fractional portions of control parameters must be kept constant at the state of '1', for example, bit b_{-1}, to ensure that the value range of control parameters in (3.56995,4.0). Therefore, the constraints make the initial values of state variables and parameters contribute the number of bits to the secret keys less than its definition.

According to the adopted values of parameters for the permutation and diffusion in this example, the number of bits represents for the secret keys is dependent on not only that of perturbed bits, but also the constraints in the value ranges of state variables and control parameters of chaotic map. Table 11 shows the number of bits for the secret keys of permutation and diffusion in different schemes of perturbation. The values of control parameters, $a_0^{(p)}$ and $a_0^{(d)}$, in the scheme of perturbation on the state variables (CPP-1 and CD-1) are fixed, so those contribute 33 and 34 bits to the secret keys, respectively, while other initial values provide 32 bits as defined by the bit patterns in Table 7. In other words, the number of bits in the secret keys can be at least 64 and 72 for the permutation and diffusion, respectively.

It is assumed that the cryptosystem consists of the permutation and diffusion with the perturbation of Logistic map as described above. Therefore, the secret key of a cryptosystem is at least 136 bits in length. That is long enough to resist from the brute force attack running on nowadays computers.

Table 11. The number of bits in the secret keys of the permutation and diffusion with perturbation.

The Number of Bits in the Secret Keys				
Scheme		**Max. No. of Bits**	**# of Bits**	**Sum**
Permutation	CPP-1	$s_{IV^{(p)}}$	32	65
		$s_{a^{(p)}}$	33	
	CPP-2	$s_{IV^{(p)}}$	32	64
		$s_{a^{(p)}}$	32	
	CPP-3	$s_{IV^{(p)}}$	32	64
		$s_{a^{(p)}}$	32	
Diffusion	CD-1	$s_{IV^{(d)}}$	32	74
		$s_{a^{(d)}}$	34	
		s_{C_0}	8	
	CD-2	$s_{IV^{(d)}}$	32	72
		$s_{a^{(d)}}$	32	
		s_{C_0}	8	
	CD-3	$s_{IV^{(d)}}$	32	72
		$s_{a^{(d)}}$	32	
		s_{C_0}	8	

5.2.5. Statistical Analyses

Here, some appropriate statistical analyses related to the content of images are carried out for the exemplar structures of permutation and diffusion. That is, the histogram, information entropy, correlation coefficients, sensitivity of secret keys, measurement of quality, and chosen-plaintext attack as well as chosen-ciphertext one are computed for this example. In the presentation of results in the tables, bad results and the best one are in italic and in bold, respectively.

It is noted that it is unbiased to compare the statistical measures for the permutation and diffusion processes in this work with those obtained by a whole cryptosystem. The simulation result is compared with that in recent works to show the advantages. However, the statistical measures for each of the permutation and diffusion show the separate contribution, if these are employed to construct a cryptosystem.

1. *Histogram analysis*

Histogram reflects the distribution of pixel values of an image. Histogram analysis of an image is considered by means of statistical histogram. The χ^2 is measured for statistical histogram. It is defined by

$$\chi^2 = \sum_{i=0}^{K-1} \frac{(O_i - E_i)^2}{E_i}, \tag{32}$$

where K is the number of grey level ($K = 256$ for 8-bit grayscale images), and O_i and E_i are respectively observed and expected occurrence frequencies of gray level i, with $0 \leq i \leq K-1$. Expected occurrence frequencies of 8-bit grayscale images is $E_i = \frac{M \times N}{K}$; M and N are the number of rows and columns of images. The unilateral hypothesis test is to consider the significance of the histogram conforming a uniform distribution. The hypothesis test is accepted (or the histogram is uniformly distributed) if $\chi^2 \leq \chi^2_\alpha(K-1)$. In this example, the significance level $\alpha = 0.05$ is considered and $\chi^2_{0.05}(255) = 293.247$.

It is noted that the analysis of histogram is only applied to the diffusion. Four original images and two special images (Black and White images) are employed in the simulation for the histogram analysis. Table 12 shows values of χ^2 which are computed for original and diffused images for different rounds of diffusion. The χ^2 values of original images are quite large in compared with those of diffused ones. It means that the histograms of original images have clear structures. Specifically, the χ^2 values of most diffused images are less then $\chi^2_{0.05}(255)$ after the first round of diffusion, or the histograms of diffused images have uniform distributions. The diffused images of Black and White have uniform histograms from the third round of iteration. However, histogram structures still exist in the first-round diffused Black and White images. It seems that there is not much difference in χ^2-test result in different schemes of perturbation. The test results show that the diffusion process provides the histogram statistics equivalent to those produced by a whole cryptosystem for example, Reference [58].

Table 12. χ^2-test results of original and diffused images.

Perturbation	Round	χ^2 Test					
		Lena	**Cameraman**	**House**	**Peppers**	**Black**	**White**
Plaintext		30,577.703	161,271.875	299,789.226	36,777.515	16,711,680	16,711,680
On state variable	1	227.977	*313.219*	*316.805*	249.102	*22,864.141*	*27,165.805*
	2	221.000	266.859	*315.852*	251.000	*340.109*	*333.063*
	3	284.180	264.977	273.344	276.367	253.516	259.984
On control parameter	1	284.086	*299.234*	*295.859*	*299.328*	*11,590.945*	*13,372.102*
	2	*299.852*	258.188	286.273	254.219	*308.008*	277.750
	3	**202.352**	241.664	270.242	253.891	238.492	286.703
On both	1	245.086	*402.266*	218.141	220.531	*20,335.578*	*27,947.445*
	2	274.539	278.383	237.500	245.391	*346.742*	*406.969*
	3	249.180	218.383	263.602	231.000	254.031	282.359

2. *Information entropy*

The information entropy $IE(V)$ is used for measuring the probability of appearance of symbol v_i in the message source V [70]. Here, the message source is the encrypted images and symbols are pixels. Calculation of $IE(V)$ for an image is

$$IE(V) = \sum_{i=0}^{2^{k_2}-1} p(v_i) log_2 \frac{1}{p(v_i)}, \tag{33}$$

where $p(v_i)$ is the probability in finding pixels with value of v_i in an image. $IE(V)$ is in bit. In the case of 8-bit grayscale image, the maximum of $IE(v_i)$ is 8 as the ideal value. Here, the entropy is only considered for diffused images only, because the permutation does not change values of pixels. Under a cryptographic point of view, the better the statistical property in a diffused image, the closer the value of $IE(V)$ to the ideal one.

Table 13 presents the information entropy of diffused images obtained by different schemes of perturbation. For four test images, the entropy of original images is much less than the ideal one, while that of most diffused images is very close to the ideal one, that is, larger than 7.99 regardless to the scheme of perturbation and the number of diffusion rounds as well. However, the diffused images of Black and White have low entropy at the first round of diffusion and it increases to the ideal one at the second and third round of diffusion. The result shows that the information entropy of diffused images are equivalent to that in most previous works, for example, References [27,29,58,62].

Table 13. Information entropy of original and diffused images.

Perturbation	Round	IE					
		Lena	Cameraman	House	Peppers	Black	White
Plaintext		7.5691	6.9046	6.4971	7.3785	0	0
On state variable	1	7.9975	7.9966	7.9965	7.9973	*7.7786*	*7.7001*
	2	7.9976	7.9971	7.9965	7.9972	7.9963	7.9963
	3	7.9969	7.9971	7.9970	7.9969	7.9972	7.9971
On control parameter	1	7.9969	7.9967	7.9968	7.9967	*7.8807*	*7.8623*
	2	7.9967	7.9972	7.9969	7.9972	7.9966	7.9969
	3	7.9978	7.9973	7.9970	7.9972	7.9974	7.9968
On both	1	7.9973	7.9956	7.9976	7.9976	*7.8051*	*7.7197*
	2	7.9970	7.9969	7.9974	7.9973	7.9962	7.9955
	3	7.9973	7.9976	7.9971	7.9975	7.9972	7.9969

3. *Correlation coefficient*

The correlation coefficient (CC) among adjacent pixels reflects one of visual properties of images, and it is high in natural images. the CCs in three directions, that is, horizontal, vertical and diagonal adjacency, are measured for a specific pixel. Thus, it is expected that CCs are close to zero in encrypted images.

Here, the CCs are considered for both permuted and diffused images, and are computed on the full range of images. Tables 14–19 show the CCs of permuted, original and diffused images for four test images. Due to special content, the CCs are computed for only diffused Black and White images. The CCs are around or larger than 0.9 for four test images, and are infinity for Black and White images. Those of transformed images are relatively close to zero, and seem to be independent from chosen scheme of perturbation and from the number of rounds. In other words, the visual structure are removed in transformed images. The result of correlation coefficients is also comparable to that given in recent reports, for example, References [27,29,58,62].

Table 14. Correlation coefficients of permuted, original and diffused Lena image.

CCs of Lena Image					
	Perturbation	**Round**	**Horizontal**	**Vertical**	**Diagonal**
Permutation	On state variable	1	−0.00149	0.00281	0.00459
		2	0.00636	−0.00316	0.00186
		3	0.00104	0.00567	−0.00178
	On control parameter	1	0.00404	0.00186	−0.00447
		2	−0.00317	0.00474	0.00226
		3	0.00432	−0.00125	−0.00538
	On both	1	−0.00177	0.00019	0.00383
		2	−0.00158	−0.00042	0.00238
		3	0.00050	−0.00266	0.00627
Plaintext			0.93998	0.96934	0.91793
Diffusion	On state variable	1	0.00400	−0.00131	−0.00288
		2	−0.00260	*0.01085*	0.00013
		3	0.00598	0.00835	−0.00248
	On control parameter	1	0.00102	−0.00715	−0.00139
		2	0.00121	0.00446	0.00829
		3	0.00150	−0.00272	−0.00231
	On both	1	0.00034	0.00272	0.00070
		2	−0.00829	0.00105	−0.00458
		3	0.00211	−0.00063	−0.00040

Table 15. Correlation coefficients of permuted, original and diffused Cameraman image.

CCs of Cameraman Image					
	Perturbation	**Round**	**Horizontal**	**Vertical**	**Diagonal**
Permutation	On state variable	1	−0.00264	0.00256	−0.00015
		2	0.00304	−0.00099	−0.00230
		3	0.00295	−0.00813	−0.00334
	On control parameter	1	−0.00007	−0.00572	0.00292
		2	−0.00148	0.00059	−0.00221
		3	−0.00002	−0.00730	−0.00038
	On both	1	0.00016	0.00699	−0.00164
		2	0.00109	−0.00307	−0.00197
		3	0.00142	0.00110	−0.00177
Plaintext			0.91957	0.95494	0.89619
Diffusion	On state variable	1	−0.00179	0.00109	0.00259
		2	−0.00550	0.00317	−0.00412
		3	−0.00091	−0.00021	−0.00421
	On control parameter	1	0.00127	−0.00033	0.00351
		2	−0.00086	−0.00360	0.00576
		3	0.00621	−0.00220	0.00038
	On both	1	−0.00147	−0.00302	−0.00257
		2	0.00396	−0.00156	0.00449
		3	−0.00305	0.00035	−0.00381

Table 16. Correlation coefficients of permuted, original and diffused House image.

CCs of House Image					
	Perturbation	**Round**	**Horizontal**	**Vertical**	**Diagonal**
Permutation	On state variable	1	−0.00188	−0.00469	0.00974
		2	−0.00069	0.00607	−0.00130
		3	−0.00218	−0.00551	−0.00226
	On control parameter	1	0.00321	0.00240	−0.00824
		2	0.00351	0.00244	−0.00432
		3	−0.00174	−0.00843	−0.00204
	On both	1	−0.00305	0.00655	−0.00033
		2	−0.00494	0.00361	0.00058
		3	−0.00776	−0.00395	−0.00247
Plaintext			0.97807	0.96528	0.94835
Diffusion	On state variable	1	−0.00276	−0.00311	0.00080
		2	−0.00514	−0.00252	−0.00372
		3	0.00223	0.00283	0.00318
	On control parameter	1	−0.00158	0.00838	−0.00022
		2	0.00250	0.00110	−0.00296
		3	−0.00127	−0.00378	−0.00435
	On both	1	0.00623	0.00086	0.00006
		2	−0.00305	0.00285	0.00833
		3	0.00254	0.00117	0.00283

Table 17. Correlation coefficients of permuted, original and diffused Peppers image.

CCs of Peppers Image					
	Perturbation	**Round**	**Horizontal**	**Vertical**	**Diagonal**
Permutation	On state variable	1	0.00126	0.00560	−0.00196
		2	−0.00378	0.00150	0.00998
		3	0.00630	0.00290	−0.00124
	On control parameter	1	−0.00130	0.00101	−0.00116
		2	−0.00210	0.00167	−0.00204
		3	0.00610	0.00559	−0.00486
	On both	1	−0.00391	−0.00237	0.00564
		2	0.00463	0.00445	0.00077
		3	−0.00186	0.00124	−0.00264
Plaintext			0.94777	0.94819	0.90359
Diffusion	On state variable	1	0.00312	0.00428	0.00276
		2	−0.00706	−0.00263	−0.00587
		3	*0.01129*	0.00016	0.00548
	On control parameter	1	−0.00205	0.00658	0.00358
		2	−0.00166	0.00271	0.00156
		3	−0.00023	−0.00465	−0.00167
	On both	1	0.00629	0.00720	−0.00560
		2	−0.00117	0.00391	0.00134
		3	0.00265	−0.00378	0.00388

Table 18. Correlation coefficients of diffused Black image.

CCs of Black Image					
	Perturbation	**Round**	**Horizontal**	**Vertical**	**Diagonal**
	Plaintext		**NaN**	**NaN**	**NaN**
Diffusion	On state variable	1	*−0.01423*	*−0.01140*	0.00404
		2	0.00063	0.00347	−0.00619
		3	−0.00670	0.00135	−0.00505
	On control parameter	1	0.00736	0.00133	−0.00894
		2	0.00318	−0.00488	−0.00340
		3	−0.00170	0.00001	0.00363
	On both	1	−0.00087	0.00442	*0.01184*
		2	−0.00169	−0.00663	0.00323
		3	0.00367	−0.00358	−0.00115

Table 19. Correlation coefficients of diffused White image.

CCs of White Image					
	Perturbation	**Round**	**Horizontal**	**Vertical**	**Diagonal**
	Plaintext		**NaN**	**NaN**	**NaN**
Diffusion	On state variable	1	0.00650	0.07976	0.00418
		2	−0.00068	0.00049	−0.00469
		3	−0.00545	−0.00087	0.00173
	On control parameter	1	*0.02599*	*0.01076*	−0.00367
		2	0.00802	0.00179	−0.00064
		3	−0.00398	0.00487	0.00032
	On both	1	*0.01336*	−0.04708	0.00202
		2	0.00213	*−0.01066*	0.00210
		3	−0.00394	0.00377	−0.00268

4. *Sensitivity of secret keys*

The sensitivity of secret key is considered by means of ciphertext difference rate (CDR) as proposed in Reference [71]. The CDR is computed by

$$Cdr = \frac{Diff(C, C_1) + Diff(C, C_2)}{2M \times N} \times 100\%, \tag{34}$$

where M and N are the size of images; C is the ciphertext using the secret key K; C_1 and C_2 are ciphertexts using the secret keys $K + \Delta K$ and $K - \Delta K$, respectively; the function $Diff(A, B)$ returns the difference in the number of pixels between images A and B. The function $Diff(.)$ is

$$Diff(A, B) = \sum_{x=1}^{M} \sum_{x=1}^{N} Difp(A(x, y), B(x, y)), \tag{35}$$

where $Difp(.)$ is

$$Difp(A(x, y), B(x, y)) = \begin{cases} 1, & \text{for } A(x, y) \neq B(x, y), \\ 0, & \text{for } A(x, y) = B(x, y). \end{cases} \tag{36}$$

It is clear that the value difference in pairs of pixels is considered for the CDR. Thus, this can be used for analyzing the sensitivity of secret keys in both the permutation and diffusion for four test images, and only in the diffusion for two special images, Black and White.

Here, the secret keys are initial values of $(IV^{(p)}, a_0^{(p)})$ for the permutation, and $(IV^{(d)}, a_0^{(d)})$ for the diffusion. Thus, the sensitivity will be considered for each of components of the secret key, and $\Delta K_{name}^{(Scheme_i)}$ denotes the difference in the component *name* of the scheme $Scheme_i$. In order to demonstrate the effectiveness, only the value of a single component of the secret key is added to and subtracted from the tolerance $\Delta K_{name}^{(Scheme_i)}$ to produce C_1 and C_2 while the other values are as previously chosen for the above simulation. The smallest value is made by the lowest significant bit for $\Delta K_{name}^{(Scheme_i)}$ in different schemes of perturbation as shown in Table 20.

Table 20. The values of Δ_K for *Cdr*.

	ΔK	Amount in Binary	Value of Tolerance
Permutation	$\Delta K_{IV}^{(CPP-1)}$	0.00000000000000000000000000000001	2^{-32}
	$\Delta K_{a}^{(CPP-2)}$	0.0000000000000000000000000000000001	2^{-34}
	$\Delta K_{IV}^{(CPP-3)}$	0.00000000000000000000000000000001	2^{-32}
	$\Delta K_{a}^{(CPP-3)}$	0.0000000000000000000000000000000001	2^{-34}
Diffusion	$\Delta K_{IV}^{(CD-1)}$	0.00000000000000000000000000000001	2^{-32}
	$\Delta K_{a}^{(CD-2)}$	0.0000000000000000000000000000000001	2^{-35}
	$\Delta K_{IV}^{(CD-3)}$	0.00000000000000000000000000000001	2^{-32}
	$\Delta K_{a}^{(CD-3)}$	0.0000000000000000000000000000000001	2^{-35}
	ΔK_{C_0}	00000001	1

The simulation is carried out with four test images and two special ones, Black and White and the results are shown in Tables 21–26 for the example that Cdr_IV, Cdr_a and Cdr_C_0 are the ciphertext difference rates with a tolerance in three initial values, IV, a, and C_0, respectively. Overall, Cdr_IV, Cdr_a and Cdr_C_0 are very close to unity with smallest tolerances in each component of the secret keys for any round of diffusion and for every scheme of perturbation. Specifically, for all images, the diffusion produces very good sensitivity to secret key with *Cdr* larger than 0.994 for every scheme of perturbation. However, for four test images, for the perturbation on state variable, the sensitivity to Cdr_IV is worse at the first round of permutation than that in larger number of permutation rounds. For every scheme of perturbation, sensitivity to Cdr_a is worse at the first round of permutation than that in larger number of permutation rounds. The result is obtained with the bit arrangements as given in Table 10, and it can be improved if higher significant bits of $x_n^{(p)}$ and $x_n^{(d)}$ are avoided to generate amounts of perturbation. Here, the result of CDRs is comparable to that in Reference [27].

In addition, the sensitivity to the secret keys can be considered by means of number of pixels change rate ($NPCR$) and unified average changing intensity ($UACI$) [72,73]. These are as

$$NPCR = \frac{\sum_{x,y} D(x,y)}{M \times N} \times 100\%, \tag{37}$$

and

$$UACI = \frac{1}{N^2}\left[\sum_{x,y} \frac{c_1(x,y) - C_2(x,y)|}{255}\right] \times 100\%, \tag{38}$$

where $D(x,y) = 1$ if $C_1(x,y) = C_2(x,y)$, and $D(x,y) = 0$ if $C_1(x,y) \neq C_2(x,y)$; C_1 and C_2 are described in Equation (34). The tolerance in the secret keys is as shown in Table 20. The simulation for six test images with the permutation and diffusion using the secret keys with and without the tolerance. The resultant images are used to compute for $NPCR$ and $UACI$. It is noted that only these are inappropriate for permuted images of Black and White.

Table 21. Ciphertext difference rates of permuted and diffused Lena image.

CDRs of Lena Image					
	Perturbation	**Round**	***Cdr_IV***	***Cdr_a***	***Cdr_C*$_0$**
Permutation	On state variable	1	*81.876*	*63.679*	-
		2	92.301	*84.467*	-
		3	96.131	92.413	-
	On control parameter	1	99.384	*80.531*	-
		2	99.424	91.125	-
		3	99.439	95.380	-
	On both	1	99.177	*81.901*	-
		2	99.414	92.160	-
		3	99.427	96.044	-
Diffusion	On state variable	1	99.485	99.516	99.528
		2	99.591	99.583	99.607
		3	99.607	99.599	99.603
	On control parameter	1	99.530	99.546	99.509
		2	99.628	99.643	99.638
		3	99.617	99.609	99.622
	On both	1	99.496	99.459	99.441
		2	99.602	99.591	99.616
		3	99.640	99.615	99.605

Table 22. Ciphertext difference rates of permuted and diffused Cameraman image.

CDRs of Cameraman Image					
	Perturbation	**Round**	***Cdr_IV***	***Cdr_a***	***Cdr_C*$_0$**
Permutation	On state variable	1	*81.145*	*63.070*	-
		2	91.579	*83.722*	-
		3	95.350	91.621	-
	On control parameter	1	98.576	*79.908*	-
		2	98.669	90.435	-
		3	98.658	94.621	-
	On both	1	98.399	*81.179*	-
		2	98.618	91.403	-
		3	98.583	95.308	-
Diffusion	On state variable	1	99.506	99.501	99.485
		2	99.598	99.541	99.601
		3	99.622	99.590	99.608
	On control parameter	1	99.550	99.565	99.550
		2	99.566	99.567	99.608
		3	99.610	99.593	99.622
	On both	1	99.471	99.494	99.489
		2	99.609	99.610	99.635
		3	99.601	99.624	99.559

Table 23. Ciphertext difference rates of permuted and diffused House image.

CDRs of House Image					
	Perturbation	**Round**	***Cdr_IV***	***Cdr_a***	***Cdr_C*$_0$**
Permutation	On state variable	1	*80.491*	*62.514*	-
		2	90.806	*83.016*	-
		3	94.501	90.883	-
	On control parameter	1	97.781	*79.177*	-
		2	97.842	*89.648*	-
		3	97.790	93.845	-
	On both	1	97.582	*80.533*	-
		2	97.720	90.611	-
		3	97.836	94.540	-
Diffusion	On state variable	1	99.505	99.526	99.534
		2	99.593	99.598	99.626
		3	99.611	99.619	99.609
	On control parameter	1	99.539	99.547	99.546
		2	99.598	99.609	99.621
		3	99.615	99.603	99.630
	On both	1	99.446	99.517	99.483
		2	99.602	99.649	99.628
		3	99.609	99.601	99.616

Table 24. Ciphertext difference rates of permuted and diffused Peppers image.

CDRs of Peppers Image					
	Perturbation	**Round**	***Cdr_IV***	***Cdr_a***	***Cdr_C*$_0$**
Permutation	On state variable	1	*81844*	*63.633*	-
		2	92.255	*84.453*	-
		3	96.104	92.378	-
	On control parameter	1	99.412	*80.538*	-
		2	99.395	91.091	-
		3	99.376	95.348	-
	On both	1	99.132	*81.837*	-
		2	99.332	92.104	-
		3	99.342	95.979	-
Diffusion	On state variable	1	99.519	99.548	99.506
		2	99.621	99.611	99.635
		3	99.590	99.593	99.596
	On control parameter	1	99.539	99.532	99.550
		2	99.612	99.570	99.601
		3	99.628	99.609	99.646
	On both	1	99.525	99.506	99.492
		2	99.614	99.593	99.605
		3	99.607	99.581	99.622

Table 25. Ciphertext difference rates of diffused Black image.

CDRs of Black Image					
	Diffusion	**Round**	***Cdr_IV***	***Cdr_a***	***Cdr_C*$_0$**
Permutation	On state variable	1	99.464	99.535	99.536
		2	99.636	99.631	99.602
		3	99.636	99.612	99.629
	On control parameter	1	99.596	99.561	99.545
		2	99.629	99.605	99.577
		3	99.610	99.612	99.608
	On both	1	99.505	99.490	99.496
		2	99.621	99.611	99.612
		3	99.612	99.596	99.619

Table 26. Ciphertext difference rates of diffused White image.

CDRs of White Image					
	Diffusion	**Round**	***Cdr_IV***	***Cdr_a***	***Cdr_C*$_0$**
Permutation	On state variable	1	99.526	99.307	99.551
		2	99.584	99.601	99.602
		3	99.635	99.597	99.608
	On control parameter	1	99.601	99.532	99.548
		2	99.608	99.596	99.609
		3	99.605	99.610	99.634
	On both	1	99.652	99.487	99.495
		2	99.587	99.596	99.625
		3	99.583	99.600	99.616

Tables 27–32 demonstrated $NPCR$ and $UACI$ for the permuted and diffused images with various schemes of perturbation. Generally, for every scheme of perturbation and for test images except two special content ones (Black and White), $NPCR$ of permutation is increased with the increase of number of rounds, and it is lower than that of diffusion for every component of secret keys. $NPCR$ of diffusion is saturated and fluctuated in the range of 99.4% to 99.7% regardless of number of diffusion rounds, schemes of perturbation, and components of secret keys.

Similarity, for test images except for Black and White and for every scheme of permutation, $UACI$ of permutation is increased with the increase of number of permutation rounds, and it is lower than that of diffusion. $UACI$ of diffusion is fluctuated within the range of 31.7% to 34.7% for every scheme of perturbation and for every component of secret keys. However, $UACI$ of permutation is different for different test images, and it is better sensitivity to Δ_{IV} than to Δ_a.

The values of $NPCR$ and $UACI$ of diffusion in this work are equivalent to those of encrypted images by a whole cryptosystem in most of previous works, for example, References [27,29,58,62].

Table 27. Sensitivity to secret keys: Lena image.

Sensitivity to Secret Keys: Lena Image								
	Permutation on	**Round**	ΔK_{IV}		ΔK_a		ΔK_{C_0}	
			NPCR (%)	**UACI (%)**	**NPCR (%)**	**UACI (%)**	**NPCR (%)**	**UACI (%)**
Permutation	On state variable	1	*98.920*	*23.359*	*62.524*	*14.660*	-	-
		2	99.370	*23.433*	*83.701*	*19.769*	-	-
		3	99.393	*23.444*	*91.956*	*21.718*	-	-
	On control parameter	1	99.361	*23.383*	*61.656*	*14.564*	-	-
		2	99.437	*23.483*	*82.838*	*19.531*	-	-
		3	99.448	*23.497*	*91.330*	*21.544*	-	-
	On both	1	*98.921*	*23.393*	*64.369*	*15.141*	-	-
		2	99.406	*23.446*	*84.898*	*19.911*	-	-
		3	99.414	*23.416*	*92.648*	*21.802*	-	-
Diffusion	On state variable	1	99.474	*32.444*	99.506	*32.512*	99.532	*32.560*
		2	99.548	33.392	99.577	33.485	99.582	33.557
		3	99.585	33.326	99.622	33.474	99.606	33.394
	On control parameter	1	99.513	*32.449*	99.550	*32.376*	99.529	*32.437*
		2	99.641	33.510	99.629	33.430	99.683	33.407
		3	99.614	33.447	99.593	33.573	99.606	33.468
	On both	1	99.468	*32.089*	99.446	*31.995*	99.426	*31.887*
		2	99.609	33.470	99.579	33.375	99.612	33.586
		3	99.638	33.480	99.623	33.533	99.620	33.416

Table 28. Sensitivity to secret keys: Cameraman image.

Sensitivity to Secret Keys: Cameraman Image								
	Permutation on	**Round**	ΔK_{IV}		ΔK_a		ΔK_{C_0}	
			NPCR (%)	**UACI (%)**	**NPCR (%)**	**UACI (%)**	**NPCR (%)**	**UACI (%)**
Permutation	On state variable	1	*98.071*	*27.877*	*61.920*	*17.583*	-	-
		2	*98.602*	*27.971*	*82.889*	*23.405*	-	-
		3	*98.647*	*28.020*	*91.190*	*25.839*	-	-
	On control parameter	1	*98.555*	*27.962*	*61.220*	*17.229*	-	-
		2	*98.674*	*28.049*	*82.205*	*23.121*	-	-
		3	*98.659*	*28.053*	*90.585*	*25.709*	-	-
	On both	1	*98.160*	*27.717*	*63.721*	*18.064*	-	-
		2	*98.653*	*27.902*	*84.224*	*23.973*	-	-
		3	*98.496*	*28.072*	*91.945*	*26.125*	-	-
Diffusion	On state variable	1	99.461	*32.415*	99.487	*32.640*	99.500	*32.382*
		2	99.600	33.496	99.567	33.348	99.609	33.454
		3	99.652	33.332	99.628	33.155	99.565	33.427
	On control parameter	1	99.542	*32.622*	99.539	*32.430*	99.516	*32.513*
		2	99.554	33.576	99.545	33.477	99.628	33.544
		3	99.608	33.472	99.616	33.351	99.617	33.427
	On both	1	99.435	*31.799*	99.490	*32.057*	99.442	*32.075*
		2	99.617	33.435	99.614	33.663	99.605	33.385
		3	99.587	33.604	99.625	33.548	99.548	33.514

Table 29. Sensitivity to secret keys: House image.

Sensitivity to Secret Keys: House Image								
	Permutation on	Round	ΔK_{IV}		ΔK_a		ΔK_{C_0}	
			NPCR (%)	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)
Permutation	On state variable	1	*97.379*	*20.122*	*61.424*	*12.763*	-	-
		2	*97.783*	*20.326*	*82.204*	*17.091*	-	-
		3	*97.697*	*20.252*	*90.462*	*18.823*	-	-
	On control parameter	1	*97.836*	*20.289*	*60.628*	*12.628*	-	-
		2	*97.862*	*20.341*	*81.473*	*16.876*	-	-
		3	*97.772*	*20.339*	*89.882*	*18.700*	-	-
	On both	1	*97.325*	*20.172*	*63.226*	*13.152*	-	-
		2	*97.707*	*20.219*	*83.489*	*17.351*	-	-
		3	*97.765*	*20.353*	*91.173*	*18.894*	-	-
Diffusion	On state variable	1	99.516	*32.538*	99.526	*32.392*	99.535	*32.441*
		2	99.553	33.448	99.585	33.333	99.619	33.575
		3	99.614	33.525	99.640	33.423	99.619	33.477
	On control parameter	1	99.498	*32.336*	99.559	*32.508*	99.559	*32.399*
		2	99.574	33.368	99.631	33.329	99.619	33.619
		3	99.614	33.439	99.605	33.383	99.623	33.398
	On both	1	99.455	*32.238*	99.507	*31.965*	99.468	*32.009*
		2	99.591	33.318	99.634	33.481	99.629	33.376
		3	99.609	33.507	99.600	33.478	99.631	33.529

Table 30. Sensitivity to secret keys: Peppers image.

Sensitivity to Secret Keys: House Image								
	Permutation on	Round	ΔK_{IV}		ΔK_a		ΔK_{C_0}	
			NPCR (%)	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)
Permutation	On state variable	1	*98.917*	*23.692*	*62.494*	*14.903*	-	-
		2	99.292	*23.818*	*83.687*	*20.009*	-	-
		3	99.377	*23.869*	*91.925*	*21.990*	-	-
	On control parameter	1	99.405	*23.712*	*61.658*	*14.658*	-	-
		2	99.446	*23.858*	*82.838*	*19.715*	-	-
		3	99.379	*23.881*	*91.324*	*21.848*	-	-
	On both	1	*98.897*	*23.673*	*64.308*	*15.397*	-	-
		2	99.313	*23.774*	*84.856*	*20.233*	-	-
		3	99.353	*23.813*	*92.627*	*22.141*	-	-
Diffusion	On state variable	1	99.539	*32.535*	99.524	*32.530*	99.489	*32.251*
		2	99.599	33.483	99.591	33.446	99.655	33.359
		3	99.617	33.597	99.629	33.506	99.564	33.451
	On control parameter	1	99.536	*32.516*	99.529	*32.735*	99.541	*32.409*
		2	99.602	33.342	99.593	33.397	99.583	33.332
		3	99.651	33.562	99.616	33.392	99.641	33.527
	On both	1	99.510	*32.037*	99.493	*31.951*	99.469	*32.133*
		2	99.597	33.601	99.594	33.534	99.625	33.599
		3	99.608	33.468	99.571	33.328	99.612	33.412

Table 31. Sensitivity to secret keys: Black image.

Sensitivity to Secret Keys: House Image								
	Permutation on	Round	ΔK_{IV}		ΔK_a		ΔK_{C_0}	
			NPCR (%)	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)
Permutation	On state variable	1	99.458	33.840	99.475	34.093	99.655	34.720
		2	99.631	33.481	99.634	33.311	99.588	33.423
		3	99.658	33.527	99.628	33.464	99.635	33.485
	On control parameter	1	99.574	*32.691*	99.550	*32.792*	99.536	*32.796*
		2	99.643	33.532	99.612	33.557	99.590	33.438
		3	99.593	33.447	99.626	33.466	99.611	33.541
	On both	1	99.498	33.002	99.512	33.075	99.501	*32.825*
		2	99.603	33.436	99.609	33.444	99.608	33.431
		3	99.612	33.476	99.609	33.337	99.614	33.361

Table 32. Sensitivity to secret keys: White image.

Sensitivity to Secret Keys: House Image								
	Permutation on	Round	ΔK_{IV}		ΔK_a		ΔK_{C_0}	
			NPCR (%)	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)
Permutation	On state variable	1	99.490	33.649	99.503	32.983	99.526	33.594
		2	99.580	33.268	99.588	33.375	99.605	33.362
		3	99.619	33.449	99.614	33.587	99.611	33.418
	On control parameter	1	99.579	*32.484*	99.539	*32.620*	99.574	*32.542*
		2	99.553	33.582	99.564	33.500	99.622	33.480
		3	99.606	33.412	99.620	33.577	99.620	33.608
	On both	1	99.715	33.137	99.490	*32.962*	99.486	*32.627*
		2	99.591	33.239	99.579	33.613	99.619	33.342
		3	99.580	33.431	99.596	33.449	99.602	33.627

5. *Measurement of permutation and diffusion quality*

Here, the quality of permutation and diffusion of three schemes of perturbation in the example is measured using the test images by the Mean-Squared-Error (MSE) and Peak Signal-to-Noise Ratio ($PSNR$). Those are performed to compare the plain images P and permuted ones C as

$$MSE = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} |P(x,y) - C(x,y)|^2, \tag{39}$$

where, $P(x,y)$ and $C(x,y)$ are values of pixels at (x,y) in P and C, respectively, and

$$PSNR = 20 \times log_{10} \left\{ \frac{255}{sqrt(MSE)} \right\}. \tag{40}$$

The larger value of MSE is, the higher quality of permutation is obtained. In contrast, the value of $PSNR$ is expected as small as possible. Table 33 shows the quality of permutation by means of MSE and $PSNR$ for four images excepted for Black and White. Values of MSE for the images are large, and those of $PSNR$ are small correspondingly. It means that most pixels of the plain image P are with values different from those in permuted one C, or high quality of permutation is obtained. However, the result shows that values of MSE and $PSNR$ are only unequal for different plain images, but independent from the schemes of perturbation and the number of permutation rounds.

Table 33. Quality of permutation based on MSE and $PSNR$.

Perturbation	Round	Lena		Cameraman		House		Peppers	
		MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
On state variable	1	5.498×10^3	10.729	9.292×10^3	8.450	4.180×10^3	11.919	5.658×10^3	10.604
	2	5.448×10^3	10.769	9.420×10^3	8.390	4.208×10^3	11.890	5.634×10^3	10.622
	3	5.475×10^3	10.747	9.359×10^3	8.419	4.200×10^3	11.898	5.636×10^3	10.621
On control parameter	1	5.471×10^3	10.750	9.422×10^3	8.389	4.184×10^3	11.915	5.679×10^3	10.588
	2	5.461×10^3	10.758	9.355×10^3	8.421	4.220×10^3	11.877	5.642×10^3	10.617
	3	5.481×10^3	10.742	9.446×10^3	8.378	4.235×10^3	11.862	5.666×10^3	10.598
On both	1	5.457×10^3	10.762	9.361×10^3	8.418	4.164×10^3	11.935	5.648×10^3	10.611
	2	5.439×10^3	10.776	9.398×10^3	8.401	4.181×10^3	11.918	5.641×10^3	10.617
	3	5.456×10^3	10.762	9.409×10^3	8.395	4.243×10^3	11.854	5.642×10^3	10.616

Besides, both MSE and $PSNR$ are also used for measuring the quality of diffusion. In addition, the sensitivity to the plain images and diffused ones is characterized the quality of diffusion by means of $NPCR$ and $UACI$. These are considered as follows. A pair of plain images, P and P_1 are diffused, in which P_1 is a modified version of P with a small change by the state of the least significant bit (LSB). The corresponding pair of diffused images C and C_1 are obtained for analyzing the sensitivity to the plaintext. Similarly, the image C' is achieved by modifying the diffused image C, and then inversely diffused to obtain the recovered plain image P'. Here, the diffusion and inverse diffusion processes are carried out on sequential pixels, therefore, the modification is made to the first pixels of P and C'. Here, the $NPCR$ and $UACI$ are given in Equations (37) and (38), and computed on the pairs of (C, C_1) and (P, P') for analyzing the sensitivity to plaintext and ciphertext.

Tables 34 and 35 display the MSE, $PSNR$, $NPCR$ and $UACI$ calculated for six pairs of test images, that is, (C, C_1) and (P, P'), to measure the quality of diffusion and inverse diffusion. Clearly, large values of MSE, $NPCR$ and $UACI$, and small values of $PSNR$ are obtained. It means that with small tolerances in P and C generate huge difference in C_1 and P', respectively; or high quality of diffusion is achieved. Overall, all of measures are independent from the schemes of perturbation and the number of diffusion rounds.

In detail, values of MSE and $PSNR$ of diffusion in Table 34 are dependent on the content of plain images, while those of $NPCR$ and $UACI$ are not. Values of MSE and $PSNR$ of Cameraman, Black and White images in the diffusion is better than those of Lena, House and Peppers images.

As given in Table 35 for the inverse diffusion, values of not only MSE, $PSNR$, but also $UACI$ are dependent on the content of plain images, and those measures of Cameraman, Black and White images are larger than those of Lena, House and Peppers. Values of $UACI$ of Black and White images are extremely good, while those of House are worse.

The quality for each of permutation and diffusion processes in this example is better than those in recent works, for example, Reference [74,75].

Table 34. Quality of diffusion based on *MSE*, *PSNR*, *NPCR* and *UACI*.

Perturbation	Round	Lena				Cameraman				House			
		MSE	**PSNR**	**NPCR**	**UACI**	**MSE**	**PSNR**	**NPCR**	**UACI**	**MSE**	**PSNR**	**NPCR**	**UACI**
On state variable	1	9.053×10^3	8.563	99.669	33.507	1.171×10^4	7.447	99.530	32.436	7.673×10^3	9.281	99.519	32.343
	2	8.950×10^3	8.613	99.582	33.446	1.174×10^4	7.435	99.640	33.405	7.720×10^3	9.255	99.611	33.529
	3	9.008×10^3	8.585	99.596	33.343	1.174×10^4	7.436	99.612	33.423	7.734×10^3	9.247	99.634	33.464
On control parameter	1	9.017×10^3	8.580	99.626	33.515	1.181×10^4	7.408	99.538	32.625	7.617×10^3	9.313	99.599	32.417
	2	9.028×10^3	8.575	99.603	33.475	1.180×10^4	7.411	99.651	33.663	7.716×10^3	9.257	99.544	33.480
	3	9.027×10^3	8.575	99.609	33.412	1.178×10^4	7.420	99.594	33.377	7.641×10^3	9.299	99.596	33.475
On both	1	9.036×10^3	8.571	99.602	33.471	1.172×10^4	7.443	99.458	32.003	7.635×10^3	9.303	99.486	32.144
	2	9.049×10^3	8.565	99.577	33.485	1.159×10^4	7.490	99.597	33.603	7.750×10^3	9.238	99.617	33.537
	3	9.078×10^3	8.551	99.596	33.557	1.173×10^4	7.438	99.637	33.425	7.706×10^3	9.262	99.658	33.567
Perturbation	**Round**	**Peppers**				**Black**				**White**			
		MSE	**PSNR**	**NPCR**	**UACI**	**MSE**	**PSNR**	**NPCR**	**UACI**	**MSE**	**PSNR**	**NPCR**	**UACI**
On state variable	1	8.388×10^3	8.894	99.539	32.344	2.842×10^4	3.595	99.638	34.519	2.783×10^4	3.685	99.550	33.651
	2	8.363×10^3	8.907	99.608	33.370	2.160×10^4	4.785	99.620	33.408	2.158×10^4	4.790	99.608	33.395
	3	8.304×10^3	8.938	99.614	33.498	2.174×10^4	4.758	99.616	33.457	2.177×10^4	4.753	99.616	33.535
On control parameter	1	8.331×10^3	8.924	99.545	32.445	2.690×10^4	3.834	99.556	32.726	2.692×10^4	3.829	99.571	32.695
	2	8.307×10^3	8.936	99.605	33.588	2.157×10^4	4.791	99.625	33.424	2.175×10^4	4.756	99.608	33.538
	3	8.292×10^3	8.944	99.585	33.350	2.172×10^4	4.762	99.622	33.489	2.173×10^4	4.761	99.631	33.366
On both	1	8.400×10^3	8.888	99.492	32.118	2.797×10^4	3.663	99.464	32.788	2.844×10^4	3.592	99.567	32.127
	2	8.317×10^3	8.931	99.645	33.318	2.162×10^4	4.781	99.585	33.380	2.188×10^4	4.730	99.634	33.577
	3	8.337×10^3	8.921	99.605	33.522	2.178×10^4	4.751	99.619	33.314	2.168×10^4	4.771	99.583	33.539

Table 35. Quality of inverse diffusion based on *MSE*, *PSNR*, *NPCR* and *UACI*.

Perturbation	Round	Lena				Cameraman				House			
		MSE	**PSNR**	**NPCR**	**UACI**	**MSE**	**PSNR**	**NPCR**	**UACI**	**MSE**	**PSNR**	**NPCR**	**UACI**
On state variable	1	9.006×10^3	8.586	99.640	30.508	1.174×10^4	7.436	99.526	34.751	7.766×10^3	9.229	99.507	28.680
	2	9.076×10^3	8.552	99.616	30.638	1.174×10^4	7.433	99.606	34.740	7.661×10^3	9.288	99.588	28.407
	3	9.033×10^3	8.573	99.612	30.610	1.168×10^4	7.457	99.600	34.599	7.652×10^3	9.293	99.567	28.457
On control parameter	1	9.064×10^3	8.558	99.602	30.650	1.172×10^4	7.440	99.527	34.648	7.758×10^3	9.233	99.542	28.635
	2	8.993×10^3	8.592	99.617	30.439	1.166×10^4	7.465	99.628	34.646	7.738×10^3	9.244	99.660	28.643
	3	8.991×10^3	8.593	99.611	30.463	1.168×10^4	7.457	99.619	34.590	7.685×10^3	9.275	99.643	28.482
On both	1	9.038×10^3	8.570	99.593	30.608	1.168×10^4	7.456	99.484	34.607	7.680×10^3	9.277	99.497	28.482
	2	9.027×10^3	8.575	99.614	30.560	1.164×10^4	7.473	99.602	34.547	7.683×10^3	9.275	99.641	28.476
	3	9.112×10^3	8.535	99.611	30.758	1.175×10^4	7.431	99.614	34.712	7.687×10^3	9.273	99.596	28.486
Perturbation	**Round**	**Peppers**				**Black**				**White**			
		MSE	**PSNR**	**NPCR**	**UACI**	**MSE**	**PSNR**	**NPCR**	**UACI**	**MSE**	**PSNR**	**NPCR**	**UACI**
On state variable	1	8.267×10^3	8.958	99.599	29.396	2.167×10^4	4.772	99.515	49.902	2.164×10^4	4.778	99.469	49.836
	2	8.338×10^3	8.920	99.521	29.474	2.155×10^4	4.796	99.583	49.718	2.179×10^4	4.748	99.634	50.126
	3	8.304×10^3	8.938	99.583	29.420	2.170×10^4	4.767	99.593	49.954	2.174×10^4	4.758	99.609	49.990
On control parameter	1	8.321×10^3	8.929	99.583	29.458	2.159×10^4	4.788	99.593	49.761	2.159×10^4	4.788	99.545	49.853
	2	8.305×10^3	8.937	99.600	29.470	2.170×10^4	4.767	99.603	49.964	2.163×10^4	4.779	99.643	49.900
	3	8.302×10^3	8.939	99.622	29.465	2.173×10^4	4.761	99.640	50.021	2.178×10^4	4.750	99.619	50.060
On both	1	8.304×10^3	8.938	99.498	29.425	2.172×10^4	4.762	99.492	50.025	2.148×10^4	4.810	99.469	49.646
	2	8.298×10^3	8.941	99.599	29.396	2.159×10^4	4.789	99.626	49.821	2.173×10^4	4.760	99.631	50.057
	3	8.322×10^3	8.928	99.643	29.464	2.170×10^4	4.766	99.611	49.968	2.175×10^4	4.756	99.652	50.046

6. *Chosen-plaintext and chosen-ciphertext attacks*

In this work, the permutation and diffusion processes are considered separately. According to the structure of perturbation as given in Section 4.1 and the figures therein, the permuted image in does not depends on the content of plain image. In other words, the permutation algorithms can not resist again chosen-plaintext and chosen-ciphertext attacks. However, the permutation process usually combines with a diffusion one in construction of cryptosystem.

Here, the diffusion algorithms as described in in Sections 4.3 and 4.4 have image-content sensitivity. The value of pixels are perturbed on the state variables and control parameters of chaotic map. This is similar to the case of authentication as given in References [57,76,77], where the hashed keys with limited lengths (e.g., 256 bits) are computed using the content of image. However, the better advantage in the proposed models in compared with previous works is that the the diffused image is dependent on every value of pixels, or it means that the length of hashed keys is equal to that of image in bits, that is, $M \times N \times k_2$ bits. Consequently, the diffusion algorithms strongly resist from the types of chosen-plaintext and chosen-ciphertext attacks.

The simulation result in this example in Tables 34 and 35 shows the image-content sensitivity by means of MSE, $PSNR$, $NPCR$ and $UACI$ as the evidence of the image-content sensitivity and resistance from chosen-plaintext and chosen-ciphertext attacks.

6. Concluding Remarks

The present work has proposed the structural models of image permutation and diffusion based on perturbed digital chaos. Dynamics of chaotic map is nonstationary during encryption. This introduces a class of chaotic ciphers utilizing the perturbation. To demonstrate the feasibility of the proposed models, the example employed the simplest chaotic map, that is, Logistic map. The simulation results of permutation and diffusion have been analyzed separately. Overall, the best result is obtained in the case of perturbation on both state variable and control parameter. The results are comparable to those reported in recent works, for example, References [27,55] and References [27,29,58,62]. There are some remarks in the proposed models of permutation and diffusion with the perturbed chaos.

Due to the dependency of image content, it should be ensured in any specific design that dynamics of chaos has good statistical properties and the cryptographic performance is obtained for special image contents. In fact, any chaotic map can be employed for the proposed models. A requirement for implementation is that the total number of perturbed bits in state variables or control parameters in a specific scheme of perturbation must be equal or larger than that representing for the coordinates and values of pixels. In addition, the key space of the proposed schemes is dependent on the number of perturbed bits. This can be expanded with the increase in the number of bits represented for state variables and control parameters in appropriate scheme of perturbation. It also means that the period of dynamics is lengthened. Besides, bits with fixed states in the value of state variables and control parameters will make value ranges of state variables and control parameters valid in separate intervals. The number of bits representing for chaotic variables and control parameters should be chosen to keep balanced between the expected size of key space and the resource available in the implementation platform.

Moreover, the structure of permutation is almost similar to that of diffusion in the same scheme of perturbation. The main difference in the structures is the way that the coordinate and the value of pixels are perturbed on state variables and control parameters, and in their recovery processes from the state variables. In the proposed structures, the rule of perturbation by means of controlling the switching is defined by Equations (19) and (18) for the permutation and by Equations (26) and (25) for diffusion. This can be changed to have better security performance. For specific sizes of images, the modulo operation can be used to figure out new coordinate of pixels in the case that the size of images along any axis is unequal to 2^n; n is an integer.

Lastly, the required resource for hardware implementation is quite low in compared with typical FPGAs. In addition, there is no operation of comparison in the hardware, thus these models can have high speed operation. Further speed can be improved by combining more than one coordinate or value of pixels perturbing on chaotic dynamics at a time. This is allowed in the case the number of perturbed bits is large enough to attain that of bits of coordinates or values of pixels. The models can be simply realized in hardware with the use of multipliers, adders, XOR gates and switches. Hardware design will be implemented on FPGAs as the future work of the proposed models.

Author Contributions: Funding acquisition, T.M.H.; Validation, S.E.A.; Writing–original draft, T.M.H.; Writing–review & editing, T.M.H. All authors have read and agreed to the published version of the manuscript.

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

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

References

1. Strogatz, S. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*; Westview Press: Boulder, CO, USA, 2015.
2. Tam, W.M.; Francis, C.M.; Lau, C.K.T. *Digital Communications with Chaos: Multiple Access Techniques and Performance*; Elsevier: Amsterdam, The Netherlands, 2007.
3. Kocarev, L.; Lian, S. *Chaos-Based Cryptography*; Springer: Berlin/Heidelberg, Germany, 2011.
4. Kumar, M.; Saxena, A.; Vuppala, S.S. A Survey on Chaos Based Image Encryption Techniques. In *Multimedia Security Using Chaotic Maps: Principles and Methodologies*; Springer International Publishing: Cham, Switzerland, 2020; pp. 1–26. [CrossRef]
5. Moafimadani, S.; Chen, Y.; Tang, C. A New Algorithm for Medical Color Images Encryption Using Chaotic Systems. *Entropy* **2019**, *21*, 577. [CrossRef]
6. Özkaynak, F. Brief review on application of nonlinear dynamics in image encryption. *Nonlinear Dyn.* **2018**, *92*, 305–313. [CrossRef]
7. Battikh, D.; Assad, S.E.; Hoang, T.; Bakhache, B.; Deforges, O.; Khalil, M. Comparative Study of Three Steganographic Methods Using a Chaotic System and Their Universal Steganalysis Based on Three Feature Vectors. *Entropy* **2019**, *21*, 748. [CrossRef]
8. Wang, L.; Cheng, H. Pseudo-Random Number Generator Based on Logistic Chaotic System. *Entropy* **2019**, *21*, 960. [CrossRef]
9. Chai, X.; Yang, K.; Gan, Z. A new chaos-based image encryption algorithm with dynamic key selection mechanisms. *Multimed. Tools Appl.* **2017**, *76*, 9907–9927. [CrossRef]
10. Saito, A.; Yamaguchi, A. Pseudorandom number generation using chaotic true orbits of the Bernoulli map. *Chaos Interdiscip. J. Nonlinear Sci.* **2016**, *26*, 063122. [CrossRef]
11. Liu, L.; Miao, S.; Hu, H.; Deng, Y. Pseudorandom bit generator based on non-stationary Logistic maps. *IET Inf. Secur.* **2016**, *10*, 87–94. [CrossRef]
12. Azzaz, M.S.; Tanougast, C.; Sadoudi, S.; Bouridane, A.; Dandache, A. FPGA implementation of new real-time image encryption based switching chaotic systems. In Proceedings of the IET Irish Signals and Systems Conference (ISSC 2009), Dublin, Ireland, 10–11 June 2009; pp. 1–6.
13. Hue, T.T.K.; Lam, C.V.; Hoang, T.M.; Al Assad, S. Implementation of secure SPN chaos-based cryptosystem on FPGA. In Proceedings of the 2012 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Saigon, Vietnam, 12–15 December 2012; pp. 129–134. [CrossRef]
14. Barakat, M.L.; Mansingka, A.S.; Radwan, A.G.; Salama, K.N. Generalized Hardware Post-processing Technique for Chaos-Based Pseudorandom Number Generators. *ETRI J.* **2013**, *35*, 448–458. [CrossRef]
15. Azzaz, M.S.; Krimil, M.A.; Labiod, F.; Kadir, A.; Teguig, D. FPGA Hardware Design of a Unified Chaotic System for CTRNG. In Proceedings of the 2018 International Conference on Signal, Image, Vision and their Applications (SIVA), Guelma, Algeria, 25–28 November 2018; pp. 1–4.
16. Kocarev, L.; Szczepanski, J.; Amigo, J.; Tomovski, I. Discrete Chaos-I: Theory. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2006**, *53*, 1300–1309. [CrossRef]

17. Oteo, J.A.; Ros, J. Double precision errors in the logistic map: Statistical study and dynamical interpretation. *Phys. Rev. E* **2007**, *76*, 036214. [CrossRef]
18. Galias, Z. The Dangers of Rounding Errors for Simulations and Analysis of Nonlinear Circuits and Systems? And How to Avoid Them. *IEEE Circuits Syst. Mag.* **2013**, *13*, 35–52. [CrossRef]
19. Li, C.; Feng, B.; Li, S.; Kurths, J.; Chen, G. Dynamic Analysis of Digital Chaotic Maps via State-Mapping Networks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2019**, *66*, 2322–2335. [CrossRef]
20. Liu, L.; Lin, J.; Miao, S.; Liu, B. A Double Perturbation Method for Reducing Dynamical Degradation of the Digital Baker Map. *Int. J. Bifurc. Chaos* **2017**, *27*, 1750103. [CrossRef]
21. Xiao, J.T.; Wang, Z.; Zhang, M.; Liu, Y.; Xu, H.; Ma, J. An image encryption algorithm based on the perturbed high-dimensional chaotic map. *Nonlinear Dyn.* **2015**, *80*, 1493–1508.
22. Assad, S.E.; Farajallah, M. A new chaos-based image encryption system. *Signal Process. Image Commun.* **2016**, *41*, 144–157. [CrossRef]
23. Shannon, C.E. *A Mathematical Theory of Cryptography*; Bell System Technical Memo MM; Alcatel-Lucent: Paris, France, 1945.
24. Preneel, B.; Rijmen, V.; Bosselaers, A. Principles and performance of cryptographic algorithms. *Dr. Dobbs J.* **1998**, *23*, 126–131.
25. Robshaw, M. *Block Ciphers*; Techreport; RSA Laboratories Technical Report TR-601; RSA Laboratories: Redwood City, CA, USA, 1995.
26. Fridrich, J. Symmetric Ciphers Based on Two-Dimensional Chaotic Maps. *Int. J. Bifurc. Chaos* **1998**, *8*, 1259–1284. [CrossRef]
27. Yavuz, E. A novel chaotic image encryption algorithm based on content-sensitive dynamic function switching scheme. *Opt. Laser Technol.* **2019**, *114*, 224–239. [CrossRef]
28. Ali, T.S.; Ali, R. A new chaos based color image encryption algorithm using permutation substitution and Boolean operation. In *Multimedia Tools and Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1–21. [CrossRef]
29. Arah, M.; Guesmi, R.; Kachouri, A. A new design of cryptosystem based on S-box and chaotic permutation. *Multimed. Tools Appl.* **2020**. [CrossRef]
30. Mastan, J.M.K.; Sathishkumar, G.A.; Bagan, K.B. A Color Image Encryption Technique Based on a Substitution-Permutation Network. In *Advances in Computing and Communications*; Abraham, A., Mauri, J.L., Buford, J.F., Suzuki, J., Thampi, S.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 524–533.
31. Panduranga, H.; Kumar, S.N. Kiran Image encryption based on permutation-substitution using chaotic map and Latin Square Image Cipher. *Eur. Phys. J. Spec. Top.* **2014**, *223*, 1663–1677. [CrossRef]
32. Solak, E.; Çokal, C.; Yildiz, O.T.; Biyikoğlu, T. Cryptanalysis of Fridrich's chaotic image encryption. *Int. J. Bifurc. Chaos* **2010**, *20*, 1405–1413. [CrossRef]
33. Li, C.; Li, S.; Lo, K.T. Breaking a modified substitution–diffusion image cipher based on chaotic standard and logistic maps. *Commun. Nonlinear Sci. Numer. Simul.* **2011**, *16*, 837–843. [CrossRef]
34. Hu, G.; Xiao, D.; Wang, Y. Cryptanalysis of a chaotic image cipher using Latin square-based confusion and diffusion. *Nonlinear Dyn.* **2017**, *88*, 1305–1316. [CrossRef]
35. Xie, E.Y.; Li, C.; Yu, S.; Lü, J. On the cryptanalysis of Fridrich's chaotic image encryption scheme. *Signal Process.* **2017**, *132*, 150–154. [CrossRef]
36. Li, M.; Guo, Y.; Huang, J.; Li, Y. Cryptanalysis of a chaotic image encryption scheme based on permutation-diffusion structure. *Signal Process. Image Commun.* **2018**, *62*, 164–172. [CrossRef]
37. Hoang, T.M.; Thanh, H.X. Cryptanalysis and security improvement for a symmetric color image encryption algorithm. *Optik* **2018**, *155*, 366–383. [CrossRef]
38. Alvarez, G.; Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurc. Chaos* **2006**, *16*, 2129–2151. [CrossRef]
39. Murillo-Escobar, M.; Meranza-Castillón, M.; López-Gutiérrez, R.; Cruz-Hernández, C. Suggested Integral Analysis for Chaos-Based Image Cryptosystems. *Entropy* **2019**, *21*, 815. [CrossRef]
40. Ye, G.; Wong, K.W. An image encryption scheme based on time-delay and hyperchaotic system. *Nonlinear Dyn.* **2013**, *71*, 259–267. [CrossRef]
41. Zhang, Y.Q.; Wang, X.Y. A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice. *Inf. Sci.* **2014**, *273*, 329–351. [CrossRef]

42. Gayathri, J.; Subashini, S. A spatiotemporal chaotic image encryption scheme based on self adaptive model and dynamic keystream fetching technique. *Multimed. Tools Appl.* **2018**, *77*, 24751–24787. [CrossRef]
43. Guo, S.; Liu, Y.; Gong, L.; Yu, W.; Gong, Y. Bit-level image cryptosystem combining 2D hyper-chaos with a modified non-adjacent spatiotemporal chaos. *Multimed. Tools Appl.* **2018**, *77*, 21109–21130. [CrossRef]
44. Cao, C.; Sun, K.; Liu, W. A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map. *Signal Process.* **2018**, *143*, 122–133. [CrossRef]
45. Zhu, S.; Wang, G.; Zhu, C. A Secure and Fast Image Encryption Scheme Based on Double Chaotic S-Boxes. *Entropy* **2019**, *21*, 790. [CrossRef]
46. Fogedby, H.; Jensen, M. Weak Noise Approach to the Logistic Map. *J. Stat. Phys.* **2005**, *121*, 759–778. [CrossRef]
47. Dettmann, C.P. Traces and determinants of strongly stochastic operators. *Phys. Rev. E* **1999**, *59*, 5231–5234. [CrossRef]
48. Palla, G.; Vattay, G.; Voros, A. Trace formula for noise corrections to trace formulas. *Phys. Rev. E* **2001**, *64*, 012104. [CrossRef]
49. Demaeyer, J.; Gaspard, P. A trace formula for activated escape in noisy maps. *J. Stat. Mech. Theory Exp.* **2013**, *2013*, P10026. [CrossRef]
50. Voss, H.U.; Timmer, J.; Kurths, J. Nonlinear Dynamical System Identification from Uncertain and Indirect Measurements. *Int. J. Bifurc. Chaos* **2004**, *14*, 1905–1933. [CrossRef]
51. McGoff, K.; Mukherjee, S.; Pillai, N. Statistical inference for dynamical systems: A review. *Statist. Surv.* **2015**, *9*, 209–252. [CrossRef]
52. Ionides, E.L.; Bretó, C.; King, A.A. Inference for nonlinear dynamical systems. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 18438–18443. [CrossRef] [PubMed]
53. Arroyo, D.; Alvarez, G.; Fernandez, V. A basic framework for the cryptanalysis of digital chaos-based cryptography. In Proceedings of the 2009 6th International Multi-Conference on Systems, Signals and Devices, Djerba, Tunisia, 23–26 March 2009; pp. 1–6.
54. Chen, J.X.; Zhu, Z.L.; Fu, C.; Yu, H.; Zhang, L.B. A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism. *Commun. Nonlinear Sci. Numer. Simul.* **2015**, *20*, 846–860. [CrossRef]
55. Wang, X.; Zhang, H. A color image encryption with heterogeneous bit-permutation and correlated chaos. *Opt. Commun.* **2015**, *342*, 51–60. [CrossRef]
56. Ye, G.; Pan, C.; Huang, X.; Zhao, Z.; He, J. A Chaotic Image Encryption Algorithm Based on Information Entropy. *Int. J. Bifurc. Chaos* **2018**, *28*, 1850010. [CrossRef]
57. Chai, X.; Gan, Z.; Zhang, M. A fast chaos-based image encryption scheme with a novel plain image-related swapping block permutation and block diffusion. *Multimed. Tools Appl.* **2017**, *76*, 15561–15585. [CrossRef]
58. Li, H.; Wang, Y.; Zuo, Z. Chaos-based image encryption algorithm with orbit perturbation and dynamic state variable selection mechanisms. *Opt. Lasers Eng.* **2019**, *115*, 197–207. [CrossRef]
59. Shen, Q.; Liu, W. A Novel Digital Image Encryption Algorithm Based on Orbit Variation of Phase Diagram. *Int. J. Bifurc. Chaos* **2017**, *27*, 1750204. [CrossRef]
60. Alawida, M.; Teh, J.S.; Samsudin, A.; Alshoura, W.H. An image encryption scheme based on hybridizing digital chaos and finite state machine. *Signal Process.* **2019**, *164*, 249–266. [CrossRef]
61. Murillo-Escobar, M.; Cruz-Hernández, C.; Abundiz-Pérez, F.; López-Gutiérrez, R.; Campo, O.A.D. A RGB image encryption algorithm based on total plain image characteristics and chaos. *Signal Process.* **2015**, *109*, 119–131. [CrossRef]
62. Chen, J.; Zhu, Z.; Fu, C.; Yu, H. An improved permutation-diffusion type image cipher with a chaotic orbit perturbing mechanism. *Opt. Express* **2013**, *21*, 27873–27890. [CrossRef]
63. Song, T. A Novel Digital Image Cryptosystem with Chaotic Permutation and Perturbation Mechanism. In Proceedings of the 2012 Fifth International Workshop on Chaos-fractals Theories and Applications, Dalian, China, 25 June 2012; pp. 202–206. [CrossRef]
64. Tong, X.; Cui, M. Feedback image encryption algorithm with compound chaotic stream cipher based on perturbation. *Sci. China Ser. F Inf. Sci* **2010**, *53*, 191–202. [CrossRef]
65. Liu, L.; Miao, S. An image encryption algorithm based on Baker map with varying parameter. *Multimed. Tools Appl.* **2017**, *76*, 16511–16527. [CrossRef]
66. Zhang, L.Y.; Li, C.; Wong, K.W.; Shu, S.; Chen, G. Cryptanalyzing a chaos-based image encryption algorithm using alternate structure. *J. Syst. Softw.* **2012**, *85*, 2077–2085. [CrossRef]

67. Cvitanović, P.; Dettmann, C.; Mainieri, R. Trace Formulas for Stochastic Evolution Operators: Weak Noise Perturbation Theory. *J. Stat. Phys.* **1998**, *93*, 981–999. [CrossRef]
68. Heninger, J.M.; Lippolis, D.; Cvitanović, P. Perturbation theory for the Fokker–Planck operator in chaos. *Commun. Nonlinear Sci. Numer. Simul.* **2018**, *55*, 16–28. [CrossRef]
69. Test Images Collections. Available online: http://www.hlevkin.com/06testimages.htm (accessed on 10 February 2018).
70. Shannon, C.E. Communication theory of secrecy systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715. [CrossRef]
71. Lian, S.; Sun, J.; Wang, Z. Security analysis of a chaos-based image encryption algorithm. *Phys. A Stat. Mech. Appl.* **2005**, *351*, 645–661. [CrossRef]
72. 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. [CrossRef]
73. Wu, Y.; Member, S.; Noonan, J.P.; Member, L.; Agaian, S.; Member, S. NPCR and UACI Randomness Tests for Image Encryption. *Cyber J. Multidiscip. J. Sci. Technol. J. Sel. Areas Telecommun. JSAT* **2011**, *1*, 31–38.
74. Bisht, A.; Dua, M.; Dua, S.; Jaroli, P. A Color Image Encryption Technique Based on Bit-Level Permutation and Alternate Logistic Maps. *J. Intell. Syst.* **2019**, *29*, 1246–1260. [CrossRef]
75. Mondal, B.; Singh, S.; Kumar, P. A secure image encryption scheme based on cellular automata and chaotic skew tent map. *J. Inf. Secur. Appl.* **2019**, *45*, 117–130. [CrossRef]
76. Yang, H.; Wong, K.W.; Liao, X.; Zhang, W.; Wei, P. A fast image encryption and authentication scheme based on chaotic maps. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 3507–3517. [CrossRef]
77. Liu, H.; Kadir, A.; Sun, X. Chaos-based fast colour image encryption scheme with true random number keys from environmental noise. *IET Image Process.* **2017**, *11*, 324–332. [CrossRef]