An Efficient Diffusion Approach for Chaos-based Image Encryption

Kwok-Wo Wong, Bernie Sin-Hung Kwok

Abstract—**A typical architecture of existing chaos-based image cryptosystems is composed of alternative permutation and diffusion stages. A multi-dimensional chaotic map is usually employed for image pixel permutation in the spatial domain while a one-dimensional (1D) chaotic map is used for diffusion purpose. As the latter usually involves real-valued arithmetic operations, the overall encryption speed is limited by the diffusion stage. In this paper, we propose a more efficient diffusion mechanism using simple table lookup and swapping techniques as a light-weight replacement of the 1D chaotic map. Simulation results show that at a similar security level, the proposed cryptosystem needs only one-third the encryption time of an existing cryptosystem. The effective acceleration of chaos-based image cryptosystems is thus achieved.**

I. Introduction

With the advancements of mobile communication technologies, the utilization of audio-visual information in addition to textual content becomes more prevalent than the past. Cryptographic approaches are therefore critical for secure multimedia content storage and distribution over open networks such as the Internet. Traditionally, Data Encryption Scheme (DES) or its equivalent is often employed to ensure the confidentiality of digital content. However, such encryption techniques are found unfit for digital images characterized with some intrinsic features such as high pixel correlation and redundancy [1]. In this regard, research in the area of image encryption using chaos theory has been emerged. One of the possible ways to resist statistical and differential attacks is to employ permutation and diffusion alternatively [2]. Typical chaos-based image encryption schemes [3-5] employ a multi-dimensional chaotic map for pixel permutation in the spatial domain while uses a one-dimensional (1D) chaotic map for diffusion purpose. However, a considerable amount of computation load is devoted to the real-valued computation and the consequent quantization usually required by the 1D chaotic map [6]. Such computation complexity outweighs its advantages in practical multimedia data encryption.

In order to improve the efficiency of typical chaos-based image cryptosystems, we propose a more efficient diffusion mechanism using simple table lookup and swapping techniques as a light-weight replacement of the 1D chaotic map. As inspired by Wong's approach [7], a two-dimensional (2D) lookup table is specially designed for pixel value diffusion purpose via dynamic swapping and relative indexing of table entries. As the table lookup and entry swapping operations are much efficient than real-valued arithmetic operations, the proposed diffusion mechanism leads to a substantial acceleration of existing chaos-based image cryptosystems.

The rest of this paper is organized as follows. In next section, the typical architecture of chaos-based image cryptosystems is introduced. Then the proposed diffusion algorithm is described in Section 3. Experimental results are reported in Section 4 while security analyses are addressed in Section 5. Finally, conclusions are drawn in the last section.

II. Typical Architecture of Chaos-based Image Cryptosystems

A typical architecture of existing chaos-based image cryptosystems is shown in Fig. 1. There are two stages in the chaos-based image cryptosystem. In the confusion stage, image pixels are permuted in a secret order, without any change in their values. The function of the diffusion stage is to modify the pixel values sequentially so that a tiny change in one pixel is spread out to many pixels, hopefully the whole image. To decorrelate the relationship between adjacent pixels, the confusion stage is performed *n* times, where *n* is usually larger than 1. This is followed by the diffusion stage. The whole *n*-round confusion and single-round diffusion repeat for *m* times, with *m* usually larger than 1, so as to achieve a satisfactory level of security. The parameters of the chaotic maps governing the permutation and the diffusion should better be different in different rounds. This is achieved by a round key generator with a seed secret key as input.

In Lian *et al.*'s cryptosystem [5], the confusion process solely permutes the pixel positions without changing their values. An invertible discretized 2D standard map with a random scan couple (r_x, r_y) given by Eq. (1) is employed in this stage.

$$\begin{cases} x_{k+1} = (x_k + y_k + r_x + r_y) \bmod N, \\ y_{k+1} = \left(y_k + r_y + K_c \sin \dfrac{2\pi x_{k+1}}{N} \right) \bmod N, \end{cases} \tag{1}$$

where (x_k, y_k) and (x_{k+1}, y_{k+1}) is the original and the permuted pixel position of an $N \times N$ image, respectively. The standard map parameter K_C is a positive integer.

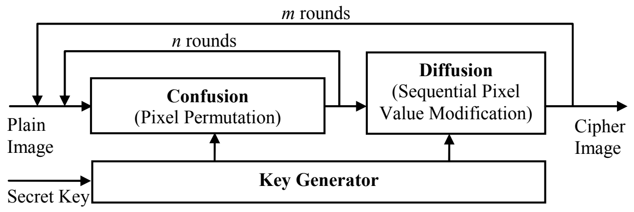

Fig. 1. Typical architecture of chaos-based image cryptosystems

The work described in this paper was fully supported by a grant from CityU [Project No. 7002070].

K. W. Wong and S. H. Kwok are with Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong; itkwwong@cityu.edu.hk (K. W. Wong)

In the diffusion stage, each pixel of the 2D permuted image is scanned sequentially, starting from the upper left corner. The diffusion effect in this stage is introduced by Eq. (2).

$$\begin{cases} c_{-1} = K_d \\ c_i = v_i \oplus q[f(c_{i-1}), L] \end{cases} \tag{2}$$

where v_i is the value of the i^{th} pixel of the permuted image, c_{i-1} and c_i is the value of the $(i\text{-}1)^{th}$ and the i^{th} pixel of the diffused image, respectively. The seed of the diffusion function is c_{-1} which is obtained from the diffusion key K_d. The nonlinear function $f(\cdot)$ is the logistic map $f(c_{i-1}) = 4c_{i-1}(1 - c_{i-1})$ while the quantization function $q(\cdot, L)$ takes the L bits immediately after the decimal point. Note that when the plain image has 256 gray levels, L is usually chosen as 8 and the logistic map input c_{i-1} is a byte value. There are only 256 possible combinations and so the logistic map output has only the same number of possible values. They are not sufficient to achieve strong diffusion.

The new pixel value is obtained by exclusive-OR (XOR) the current pixel value v_i of the permuted image with an L-bit sequence obtained from the logistic map taking the previous diffused pixel value c_{i-1} as input. As the previous diffused pixel will affect the current one, a tiny change in the plain image is reflected in more than one pixel in the cipher image and so the diffusion effect is introduced in this stage. To generate the distinct confusion and diffusion keys used in different rounds, a key generator composed of skew tent maps is suggested in [5].

III. The Proposed Algorithm

As described in the previous section, a 1D chaotic map such as the logistic map is usually employed in the diffusion stage. For security purpose, the computing precision of this 1D chaotic map must not be too low. As a result, the real-valued arithmetic operation and the consequent quantization steps consume much computation time. Inspired by Wong's approach for document encryption [7], here we propose a more efficient diffusion algorithm based on a dynamic lookup table and the two-dimensional (2D) chaotic map already computed in the permutation stage. Details of the proposed diffusion approach are described as follows.

A. Efficient Diffusion Approach Using Table Lookup and Swapping

Assuming that a gray scale plain image of size $N\times N$ and its corresponding cipher image is $P = \{P_0, P_1, \ldots, P_{N\times N-1}\}$ and $C = \{C_0, C_1, \ldots, C_{N\times N-1}\}$, respectively. Each element of P and C is an 8-bit value representing the gray level of that pixel. The procedures of the diffusion process are described as follows.

1) Construction of Tables: The proposed diffusion approach requires the lookup in two tables, a 256 × 256 table M and a 16 × 16 table T. Table M is obtained from the permutation stage, without additional computation. Like other chaotic image encryption schemes, the permutation stage of our scheme relocates each pixel to a new position uniquely by iterating a 2D chaotic map such as cat map, baker map or standard map. However, our scheme requires that the mapping information is preserved after the permutation rounds. If the image width N is larger than 256, the permutation map is larger than 256 × 256. The upper left corner of table M is determined by part of the secret key. Then 256 rows and 256 columns are obtained consecutively from the permutation map to fill table M in a cyclic manner. On the contrary, if $N < 256$, the 256 × 256 chaotic mapping positions should be computed, which is larger than the necessary size for permutation.

The 16 × 16 table T is created in the diffusion stage. Each table entry is an integer uniquely drawn from the integer set {0, 1, 2, …, 255} so as to represent all the possible 8-bit gray levels of an image pixel. The initial table arrangement is key-dependent and should be kept secret.

2) Table Lookup for Diffusion: For the encryption of the permuted pixel P_i, both the diffusion round key, the previously processed pixel P_{i-1} and its encrypted version C_{i-1} are involved. To avoid potential chosen plaintext attack and known plaintext attack, the byte values of P_{i-1} and C_{i-1} are firstly encoded by some means before encryption. A simple way is to use their values to lookup a new entry from table T. The upper 4 bits are used to identify a row in T while the lower 4 bits are taken as the column index, as defined in Eqs. (3) & (4).

$$P'_{i-1} = T(U(P_{i-1}), L(P_{i-1})), \tag{3}$$

$$C'_{i-1} = T(U(C_{i-1}), L(C_{i-1})), \tag{4}$$

where $U(\cdot)$ and $L(\cdot)$ is respectively the upper and lower 4-bit value of a byte.

The encoded values P'_{i-1} and C'_{i-1} are then used as the row and column indices to lookup an entry in table M. That entry contains a ($\log_2 N+\log_2 N$)-bit tuple. As this length is longer than 8-bit, a proper scaling is required in order to make it a proper index to table T. By the modulo-16 operation, the ($\log_2 N+\log_2 N$)-bit tuple is reduced to a (4+4)-bit one, as denoted by (x_4', y_4'). The latter is then employed to lookup a masking entry $T(x_4', y_4')$ in table T, which is finally XORed with the permuted pixel P_i to generate the cipher pixel C_i. These operations are governed by Eqs. (5) & (6).

$$(x_4', y_4') = M(P'_{-1}, C'_{i-1}) \bmod 16, \tag{5}$$

$$C_i = P_i \oplus T(x_4', y_4'). \tag{6}$$

Since the cipher pixel C_i depends on the mask which is generated by the previous plain pixel P_{i-1} and cipher pixel C_{i-1}, a small change in a pixel is most likely to influence all subsequent pixels. The diffusion effect is thus achieved. Note that when the first plain pixel P_0 is being encrypted, the initial value P_{-1} and C_{-1} is determined by the sub-key K_x and K_y, respectively.

An illustration of the proposed diffusion algorithm is given in Fig. 2. The operations required are simple logical

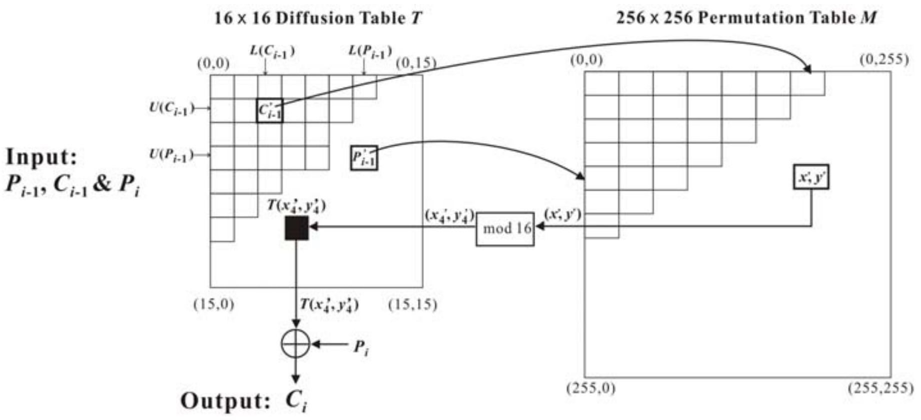

Fig. 2. Block diagram of the proposed diffusion algorithm

XOR, bitwise-AND operations and some memory load operations. All of them can be performed in high speed and so the diffusion stage becomes more efficient. Note that in the reversed procedure of pixel permutation, the chaotic map is inversely used to move permuted pixels back to the original position, whereas the map information for diffusion has to be the same in both encryption and decryption. For this reason, the information in the reversed permutation is required to remap the inversed chaotic map. This results in the allocation of an additional 256 × 256 memory space for two 8-bit inputs, P_{i-1} and C_{i-1}. This extra memory allocation has very little effect on the decryption speed. Apart from this, the decryption procedures are the same as the encryption ones but operating on the cipher image.

3) Update Table T: For security purpose, the content of table T should better keep changing throughout the diffusion process. This is realized by entry swapping and relative indexing using the values of P'_{i-1}, C'_{i-1} and (x_4', y_4'). The update is performed by swapping an entry pair (x_4', y_4') and (u, v). The value of u and v is determined by the lower 4-bit of P'_{i-1} and C'_{i-1}, respectively, i.e., $u = L(P'_{i-1})$, and $v = L(C'_{i-1})$. Note that sometimes (u, v) and (x_4', y_4') point to the same entry. In this case, the diagonal neighboring entry (u+1, v+1) will swap with (x_4', y_4') instead.

In order to enhance the above-mentioned dynamical property, the content of table T should be further changed in addition to explicit entry swapping. One of the possible methods is to cyclic shift the entire table by certain rows and columns. This further ensures that the table for encrypting the next pixel is completely different from the current one. Instead of direct table content shifting, the technique of relative indexing can be employed. This means that the relative index (r_s, r_t) corresponding to the current origin can be added to the row and column index in every table lookup operation. Initially, the value of (r_s, r_t) is derived from the sub-key K_r. Afterwards, the next relative index (r_s, r_t) can be simply assigned the location of the last mask entry (x_4',y_4'), and is therefore kept updating throughout all image pixels.

Figure 3 shows the content of table T before and after update. Suppose that the white square represents entry (u,v) while the black one corresponds to entry (x_4',y_4'). To perform the entry swapping operation, the white and black

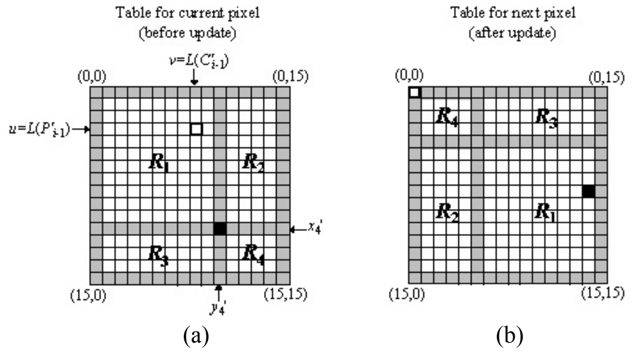

Fig. 3. An illustration of the dynamic update of lookup table T

squares are first exchanged. Then the position of the white square becomes the new origin of table T. After these two steps, the table is shown in Fig. 3(b), which is totally different from that in Fig. 3(a). Since the table update is governed by both plaintext and ciphertext, this further enhances the diffusion effect.

B. Architecture of the Proposed Cryptosystem

The overall architecture of the proposed chaos-based image cryptosystems is shown in Fig. 4. The operation procedures are described as follows:

Step 1: Generate the required sub-keys by the key generator which is triggered by a secret key in the first round. Three sub-keys are required for the random scan couple (r_x, r_y) and the standard map parameter K_C in the permutation stage. The length of both r_x and r_y are $\log_2 N$-bit. Moreover, K_C is scaled to make it in the range (5000, 7500). In the diffusion stage, three 8-bit sub-keys K_x, K_y and K_r are needed. The first two are used to determine the initial value P_{-1} and C_{-1} for masking the first permuted pixel P_0. The initial relative index (r_s, r_t) of table T is determined by the upper and lower 4-bit of K_r, respectively. If the image width N is larger than 256, two additional sub-keys K_{trow} and K_{tcol} are required to select 256×256 entries from the permutation map to form table M.

Step 2: Shuffle every element in the set $P = \{P_0, P_1, \ldots, P_{N\times N-1}\}$ for n rounds using a 2D standard map given by Eq. (1). The data set obtained is called the

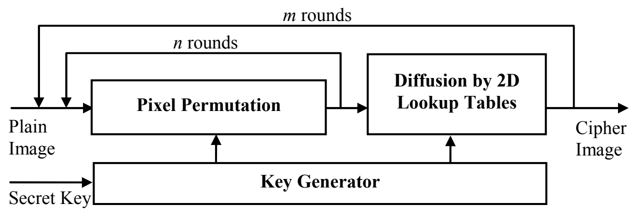

Fig. 4. Architecture of the proposed chaos-based image cryptosystem

intermediate set *I*. This permutation step also gives $N \times N$ position mapping information which can be re-used to form table *M* in the diffusion step.

Step 3: Perform the diffusion operation using 2D lookup tables *T* and *M*. Encrypt all pixels in set *I* by masking each pixel sequentially and then output the cipher image data set $C = \{C_0, C_1, \ldots, C_{N\times N-1}\}$.

Step 4: Repeat the above steps for *m* times with different sub-keys to satisfy the security requirements.

IV. Simulation Results

In this section, simulation results will be given to demonstrate the efficiency and the effectiveness of the proposed diffusion mechanism. As mentioned previously, any 2D or higher dimensional discretized chaotic maps can be employed and collaborated with the proposed diffusion approach. Following Lian *et al.*'s studies [5], a 2D standard map has a comparatively larger key space and so is adopted in the permutation process.

One of the security requirements of an effective image encryption scheme is the ability of resisting differential attacks. Two performance indices, namely, number of pixels change rate (NPCR) and unified average changing intensity (UACI) used in [3, 4] are computed to measure the influence of a plain pixel change on the entire cipher image. Table I lists these two performance indices of the proposed and Lian *et al.*'s scheme at different combinations of permutation (*n*) and overall (*m*) rounds, using the standard 512 × 512 gray-scale Lena image.

TABLE I
Encryption time, decryption time and performance indices NPCR and UACI of the proposed and Lian *et al*'s schemes, for some selected combinations of *m* and *n*

	Encryption Time (*ms*)		Decryption Time (*ms*)		NPCR		UACI	
m,n	Proposed	Lian *et al*	Proposed	Lian *et al*	Proposed	Lian *et al*	Proposed	Lian *et al*
1,2	24.20	14.80	25.16	15.35	0.37994	0.000179	0.127498	0.00004
1,3	28.67	19.39	29.75	19.78	0.066723	0.000252	0.02225	0.000061
1,4	33.01	24.13	34.28	24.27	0.769768	0.000423	0.258499	0.000093
2,1	**39.39**	20.63	**41.53**	21.73	**0.996117**	0.002464	**0.334788**	0.000486
2,2	48.52	29.62	50.55	30.77	0.996178	0.009903	0.335223	0.002623
2,3	57.30	38.69	60.19	39.77	0.995991	0.019802	0.335238	0.005082
2,4	66.22	48.04	69.02	48.75	0.996136	0.031902	0.334199	0.008362
3,1	59.14	30.93	62.63	32.55	0.99593	0.085651	0.334508	0.021158
3,2	72.58	44.64	76.62	46.14	0.996201	0.44632	0.334134	0.121025
3,3	86.07	58.15	90.37	59.84	0.995918	0.647816	0.334949	0.176647
3,4	99.29	72.02	103.49	73.17	0.996281	0.748901	0.335248	0.205962
3,5	112.77	85.43	117.29	86.71	0.996105	0.21764	0.334782	0.05957
4,1	79.21	41.52	84.44	43.42	0.995975	0.771065	0.334585	0.212241
4,2	97.00	59.29	101.72	61.43	0.996059	0.984406	0.334328	0.300348
4,3	114.95	77.58	120.46	79.96	0.996181	0.99091	0.334931	0.311426
4,4	132.26	**95.81**	138.47	**97.90**	0.996166	**0.992676**	0.334347	**0.317068**
4,5	150.05	113.69	156.31	116.16	0.996124	0.960281	0.334779	0.280655
5,1	99.19	51.59	104.84	54.66	0.99614	0.992588	0.334266	0.316712
5,2	121.90	74.35	127.90	76.94	0.996277	0.995815	0.333991	0.329292
5,3	143.65	96.88	150.52	99.56	0.996342	0.995861	0.333831	0.33097
5,4	166.03	119.69	173.33	122.39	0.996166	0.995892	0.334605	0.333018
5,5	188.16	142.25	195.46	144.78	0.996105	0.995693	0.333785	0.327371
6,1	119.18	62.24	126.34	65.45	0.996101	0.996037	0.335647	0.332429
6,2	145.87	89.50	153.65	92.45	0.996216	0.996109	0.334438	0.333748
6,3	172.29	**116.33**	180.52	**119.53**	0.996181	**0.995865**	0.333489	**0.334197**
6,4	199.64	143.53	208.58	146.72	0.995861	0.996101	0.334204	0.334517

In general, the performance is better with more cipher rounds. One can see that there is a fluctuation between the two schemes in the first overall round (m=1). Since only one diffusion step is performed, the degree of differential influence is subject to the position of the pixel change. After two overall rounds (m=2, n=1), the proposed scheme has already gained sufficiently high performance in both NPCR and UACI tests with NPCR=0.996 and UACI=0.334. They are better than Lian *et al*.'s recommendation (m=n=4) in [5] and is as good as the case (m=6, n=3). This is because more overall rounds are required in their cryptosystem to compensate for the weak diffusion.

The encryption and decryption time are measured by running a Visual C++ program in a personal computer with a Pentium D 3GHz processor, 512MB memory and 80GB hard-disk capacity. The timing results are listed in Table I as well. In general, Lian *et al*.'s cryptosystem is faster than ours under the same (m,n) choice. This is because there are only 256 possible outcomes in their diffusion stage. They are all pre-calculated and stored in a table for later lookup. As a result, it is not necessary to perform the real-valued logistic map calculation each time and much computation time is saved. At the security level of NPCR=0.996 and UACI=0.334, the proposed scheme needs 39.39 *ms* (m=2, n=1) for encryption which is only one-third the time required by Lian *et al*.'s scheme, i.e., 116.33 *ms* (m=6, n=3). In addition to programming realization factors, a slight difference in the encryption and decryption time of the proposed scheme also comes from the mapping conversion in the reversed procedure of pixel permutation. Nevertheless, the corresponding decryption time (41.53 *ms*) is only slightly longer than one third of Lian *et al*.'s (119.53 *ms*). Since the proposed diffusion approach only involves those naturally fast operations such as logical and memory load operations, it is comparatively effective than other approaches based on real-valued chaotic maps.

V. Security Analysis

In this section, the proposed chaos-based image cryptosystem is analyzed using different security measures. They include key space analysis, key sensitivity test and correlation analysis of adjacent pixels.

A. Key Space Analysis

In the proposed scheme, all initial values and system parameters in permutation and diffusion stages can be considered as part of the secret key. Assume that the key generator can generate as many different keys as required, i.e., the key space is not limited by the key generator. In the diffusion stage, the parameters are K_x, K_y, K_r, K_{trow}, K_{tcol} and the initial setting of the 16 × 16 2D diffusion table T. The worst case is that the image width N is smaller than or equal to 256. Then the values of K_{trow} and K_{tcol} are both equal to one. The corresponding key space is $2^8\cdot2^8\cdot2^8\cdot1\cdot1\cdot256!$, equivalent to 1.37×10^{515}. This key space is enlarged to $2^8\cdot2^8\cdot2^8\cdot N\cdot N\cdot256!$ when N is larger than 256. It is equal to 3.43×10^{520} when N is 512. When taking into consideration the parameters in permutation, the key space of the proposed cryptosystem would be even larger and make brute-force attack infeasible.

B. Key Sensitivity Test

We perform this test on the 512×512 Peppers image. First of all, 5 sub-keys, namely, K_x, K_y, K_r, K_{trow} and K_{tcol} are chosen. The encryption is then carried out to obtain the cipher image. Then a 1-bit difference is introduced in the chosen key and the encryption is performed again with all other parameters remain unchanged. The pixel gray scale values of the two cipher images are then compared.

Suppose that the chosen key is K_x = 123, K_y = 34, K_r = 56, K_{trow} = 78, K_{tcol} = 90 while the slightly different key is K_x = 123, K_y = 34, K_r = 56, K_{trow} = 78, K_{tcol} = 91, with the value of K_{tcol} changed from 90 to 91. The corresponding cipher image is depicted in Figs. 5(b) & 5(c), respectively. Figure 5(d) shows the difference image between them which is found 99.62% dissimilar to each other.

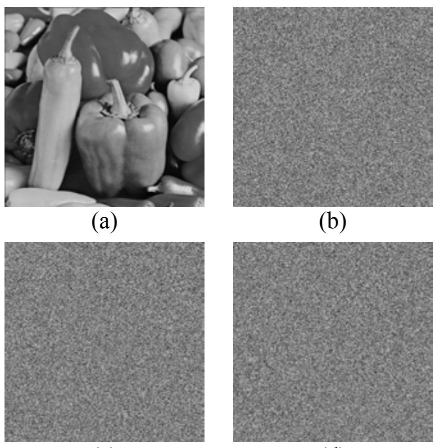

Fig. 5. Key sensitivity test 1: (a) Plain image; (b) Cipher image (K_x = 123, K_y = 34, K_r = 56, K_{trow} = 78, K_{tcol} = 90); (c) Cipher image (K_x = 123, K_y = 34, K_r = 56, K_{trow} = 78, K_{tcol} = 91); (d) gray scale difference between the two cipher images.

TABLE II
Results of Key Sensitivity Test

	Key 1 (Original value)	Key 2 (Modified value)	Pixel difference in encryption	Pixel difference in decryption
K_x	123	122	99.6128%	99.6155%
K_y	34	35	99.5899%	99.6063%
K_r	56	57	99.6044%	99.6105%
K_{trow}	78	79	99.6048%	99.6120%
K_{tcol}	90	91	99.6174%	99.6098%

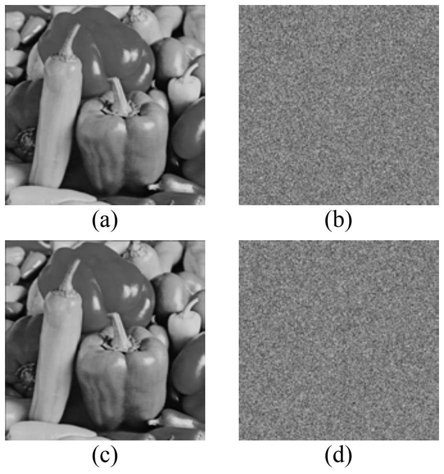

Fig. 6. Key sensitivity test 2: (a) Plain image; (b) Cipher image (K_x = 123, K_y = 34, K_r = 56, K_{trow} = 78, K_{tcol} = 90); (c) decrypted image (K_x = 123, K_y = 34, K_r = 56, K_{trow} = 78, K_{tcol} = 90); (d) decrypted image (K_x = 123, K_y = 34, K_r = 56, K_{trow} = 78, K_{tcol} = 91).

TABLE III
CORRELATION COEFFICIENTS OF TWO ADJACENT PIXELS IN THE PLAIN AND THE CIPHER IMAGES

	Plain Lena image	Cipher image
Horizontal	0.975103	0.003828
Vertical	0.988925	-0.001135
Diagonal	0.96704	0.001023

In addition to testing with slightly different encryption keys, decryption using keys with only 1-bit difference is also performed. Figure 6 shows the experimental results. As observed from the figure, the decrypted image is totally different from the plain one even when there is only a 1-bit difference in the decryption key. Similar results for other sub-keys K_x, K_y, K_r and K_{trow} are tabulated in Table II.

C. Correlation Analysis of Two Adjacent Pixels

A statistical test on the correlation between two adjacent pixels of the cipher image [3,4] has been carried out. After performing two overall cipher rounds with (m=2, n=1), 1000 pairs of adjacent pixels are randomly selected from the horizontal, vertical and diagonal directions, respectively. The correlation coefficients of the pixel pairs are then calculated and are listed in Table III, together with the results of the plain image. The correlation coefficients justify the claim that neighboring pixels of the plain image are decorrelated by the proposed cryptosystem effectively.

VI. CONCLUSIONS

In this paper, an efficient diffusion approach is proposed to address the efficiency problem in chaos-based image encryption. Our approach makes use of a static 2D table generated in the permutation stage and a dynamic 2D lookup table created in the diffusion process. The position and the value of each permuted image pixel are used to lookup the tables so as to obtain a new 8-bit value to mask the permuted pixel value. Its performance is compared with Lian *et al.*'s cryptosystem [5] using real-valued 1D chaotic map for diffusion. Experimental results show that at a similar security level, the proposed method requires fewer overall rounds and hence a shorter encryption time than those common diffusion techniques based on real-valued 1D chaotic maps. Our encryption time is only one-third of that required by Lian *et al.*'s scheme. This is because the table lookup and swapping operations are much faster than the floating-point arithmetic operations. This significant acceleration enhances the practical applications of chaos-based image cryptosystems.

REFERENCES

[1] S. Li, G. Chen, and X. Zheng, "Chaos-based encryption for digital images and videos," in Furht B and Kirovski D (Eds.), *Multimedia Security Handbook*, Ch. 4, pp.133-167, CRC Press, 2005.

[2] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcat Chaos* 8(6), pp.1259-1284, 1998.

[3] G. Chen, Y. B. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons and Fractals* 21(3), pp.749-761, 2004.

[4] Y. B. Mao, G. Chen, and S. G. Lian, "A novel fast image encryption scheme based on the 3D chaotic baker map," *Int. J. Bifurcat. Chaos* 14(10), pp.3613-3624, 2004.

[5] S. G. Lian, J. Sun, and Z. Wang, "A block cipher based on a suitable use of chaotic standard map," *Chaos, Solitons and Fractals* 26(1), pp.117-129, 2005.

[6] T. Xiang, K. W. Wong, and X. Liao, "A novel symmetrical cryptosystem based on discretized two-dimensional chaotic map," *Phys. Lett. A* 364, pp.252-258, 2007.

[7] K. W. Wong, "A fast chaotic cryptographic scheme with dynamic look-up table," *Phys. Lett. A* 298(4), pp.238-242, 2002.