Multimed Tools Appl (2017) 76:16107–16123
DOI 10.1007/s11042-016-3897-x

An improved method for digital image fragile watermarking based on chaotic maps

Mahboubeh Nazari[1] · **Amir Sharif**[2] · **Majid Mollaeefar**[2]

Received: 21 March 2016 / Revised: 6 August 2016 / Accepted: 22 August 2016 /
Published online: 15 September 2016
© Springer Science+Business Media New York 2016

Abstract In this paper, the digital image fragile watermarking method based on chaotic maps is proposed. Our method has some significant advantageous in comparison with other available methods. Firstly, we reduce watermark payloads, while they have high quality of recovery and security. In watermark embedding phase, we process the image in order to produce the information array for each block, which finally embedded in the host image to build watermarked image. The information array for each block has different length, which is defined based on block characteristic that could be smooth or rough. The second superiority of the proposed method is proposing a new metric for calculating roughness of image block, which leads to less consume of bandwidth in comparison with other available methods. Finally, we use chaotic map for block-mapping that enhances the security. Our method provides basic requirements of watermarking scheme such as, invisibility, recover quality and security. Experimental Results have proved that our method is powerful in tamper detection, self-recovery and robust against known watermarking attacks.

Keywords Fragile watermarking · Chaos · Tempering localization · Image security

✉ Mahboubeh Nazari
ma.am.math@gmail.com

Amir Sharif
amir.sharif@imamreza.ac.ir

Majid Mollaeefar
m.mollaeefar@imamreza.ac.ir

[1] Department of Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran

[2] Department of Computer and Information Technology, Imam Reza International University, Mashhad, Iran

1 Introduction

The security systems could be described in two types, the first one is 'cryptography' that consist of text encryption, image encryption [11, 13, 17, 24, 28, 29], and etc. The second one is 'information hiding', which consists of 'Steganography' and 'Watermarking'. The watermarking's purpose is authentication of digital content and recovering the original content in tempered regions by the help of embedding additional information in the host image. In order to achieve this goal, we produce an information array for each block of image and embed it in the host image. Two basic steps in watermarking are, dividing an image to the same size blocks and producing an information array for each block by processing these blocks. Some previous methods using Compression Code (*CC*), i.e. the quantized *DCT* coefficients [7, 10], *VQ* indexing [26], and average intensity [15, 20, 21, 26, 27]. In these methods, the number of bits, which are produced is fixed for all blocks. If we use a fixed size for all blocks we would face a problem, which may constraints result, because the provided information is over fit for the smooth block and inadequate for rough blocks. We are facing two challenges in the watermarking algorithms, the large information array length that increases the watermark payload and the insufficient data makes problem in recovery phase and decrease the image quality. To overcome these problems some schemes have been proposed. For instance, in [20, 21], the authors proposed a multi-level encoding for various types of blocks with variable of information array (*CC*). However, in these methods, the capacity of watermark embedding that is fixed, lead to the size fixation of *CC* for different images. "Qin et al." [22], proposed a novel reversible data hiding in the index tables of vector quantization compressed image based on index mapping mechanism. *VQ* indices with zero occurrence besides high occurrence indices are used to make a mapping list. These schemes still lead to overmuch or inadequate information array length for the smooth or the rough image blocks. In 2014, "Chen et al.". [8] proposed a method for digital image watermarking with flexible watermark payload. They used a chaotic pseudorandom number generator for encrypting the watermarked data and block mapping. They used a metric for defining block types that classified into smooth and rough. Their metric is dysfunctional, because it wrongly recognized some smooth blocks as rough. This shortcoming results consumption of bandwidth in improper way, because they send 12 bits instead of 6 bits for smooth blocks. Other disadvantages of this method, for recovering a tampered block, they employed an average mapping block in the valid case, otherwise average of 8 neighbor blocks are utilized for replacing the whole pixels in tampered block. This method for recovery may have a convincing results, but it could be better by using features of valid neighbor block pixels for determining each pixel in this tampered block. In 2015, "Liao and Shu" [16], proposed a novel method for reversible data hiding in encrypted image based on absolute mean difference of multiple neighboring pixels. They suggest a new metric to calculate complexity of image blocks that leads to high data embedding ratio. "Chen et al." [6], proposed color image analysis based on Quaternion-Type-Moments. Indeed, a novel *QTMs* and *QTM* invariants (*QTMIs*) are suggested which had better performance in several application frameworks such as: color image reconstruction, face recognition, and etc. In this paper, we proposed a fragile digital image watermarking method by using chaotic map, which has a better definition for classification of smooth and rough blocks and better quality in self-recovery phase. The original image is divided into equal 2×2 sub-blocks, and then we categorize the blocks into rough or smooth blocks. After that, based on the block's type, we create information array with size of 6 for smooth or 12 for rough blocks. In order to enhance security, we use a chaotic map for block mapping and after encrypting the watermarked data

with a secret key, we embed the data in position, which defined by block mapping procedure. Moreover, we suggest a novel strategy for self-recovery of each pixel in tampered block based on position of tampered blocks in image and features of valid neighbor block pixels. Experimental results proved that our scheme is secure and has a good quality in self-recovery phase.

The paper structure is as follows: in section 2, the proposed method is described. Then we provided experimental results in section 3 and in section 4, we come into the conclusion.

2 The proposed method

The proposed method consists of 4 phases (block encoding, watermark embedding, watermark extraction and verification and image recovery), which separately define in following sections. The overall views of each phase in proposed scheme are shown in Figs. 1, 2, 3, and 4, respectively.

2.1 Block encoding

In this phase, first we divide the original Image test (I), into N, non-overlap sub-blocks $I_1 \ldots I_N$ with size of 2 × 2 then, for each sub-block I_i, we extract five most significant bits of each pixel to form a matrix A_i. The information array contains of 5-bits average, 1-bit block type and 6-bits that are added when the block is rough. With these 6 bits we can determine exact position of maximum pixel and the positions of pixel(s) that are less than average. This information is very vital in self-recovery phase of tampered region. The information array corresponding to block I_i denoted by (α_1). The overview of information array block is shown in Fig. 5. The detail steps of block encoding are as follows:

Step 1: Average-code: For each 2 × 2 blocks, let $A_i = \begin{pmatrix} a_{i1} & a_{i2} \\ a_{i3} & a_{i4} \end{pmatrix}$, is an average matrix corresponding to sub-block I_i that has integer entries between (0, 31), where a_{ik} (k = 1, 2, 3, 4) is five most significant bits of corresponding pixels in sub-block I_i.

The average value of the block is: $Avg_i = floor\left(\frac{1}{4}\sum_{j=1}^{4} a_{ij}\right)$ Where floor, rounds each argument to the nearest integer less than or equal to that argument, and the average belong to [0, 31] interval. We define an array that called high frequency array (H) as follows:

$$H = \begin{pmatrix} a_{i1} - Avg_i & a_{i2} - Avg_i \\ a_{i3} - Avg_i & a_{i4} - Avg_i \end{pmatrix}$$

The aforementioned array can contain one, two, or three negative element(s), which their average use in definition of parameter S that will discuss in following.

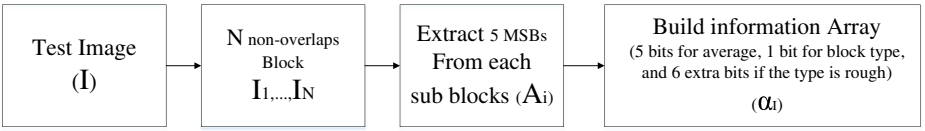

Fig. 1 The overview of block encoding process

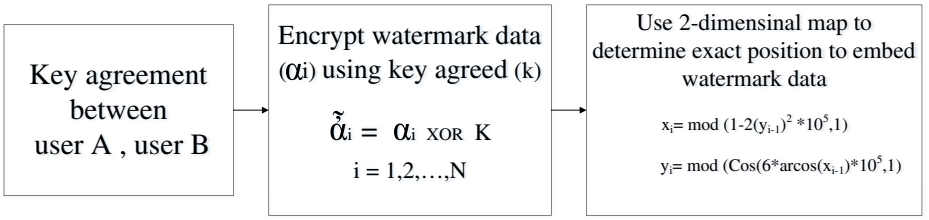

Fig. 2 The overview of watermark embedding process

Step 2: Block-type: We use the Eq. (1), as stated bellow, in order to classify the block as smooth or rough.

$$Block\ type_i = Max_i - (Avg_i + |sum_{H_i}|) \tag{1}$$

Where Max_i and sum_{H_i} demonstrate the maximum element of matrix A_i and the sum of elements of H_i, which are negative, respectively. If the above equation result become negative, the block is rough, otherwise the block is smooth. This new metric provides a strong tool for better quality in self-recovery phase and efficient use of bandwidth. Table 1, shows the typical expression for certain 2×2 blocks. The average intensity and *CC* are also shown. As it is obvious base on Table 1 (the highlighted row), the proposed metric in [8] is inefficient due to the fact its distinguishes an smooth block as rough block, which lead to send 12 bits instead of 6 bits and further bandwidth consumption.

Step 3: We fulfill positions of 7–12 in information array, if only the block is rough. These bits are filled based on Tables 2 and 3 and the rule described in Eq. 2.

$$S = \begin{cases} 0 & if\ \ 2MSBs\ \ Max\ \ pixel=''11''\ \ and\ \ 2Msbs\ \ of\ \ average\ \ of\ \ negative\ \ elements\ \ in\ \ H\ \ array=''00'' \\ 1 & otherwise \end{cases} \tag{2}$$

For example, status 3 in Table 1 demonstrate that maximum element placed in second position in A_i array (a_{i2}). The *2MSBs* of maximum pixel and *2MSBs* of average of negative element(s) in H array equals to "11", "00", respectively. For this condition, *S* is selected as zero, which mean that there exist a big difference between maximum pixel and other pixel(s) that have negative value(s) in *H* array. Indeed, such condition will occur in edge of the image.

By excluding the position of maximum element of matrix A_i, three positions remain that denoted by numbers 1, 2, and 3 in above table relative to maximum. For example, status 4 in above table means that position of negative elements relative to maximum are first and second place. The tampered region in edges could be recovered with better quality in our scheme based on better definition of information array in our scheme.

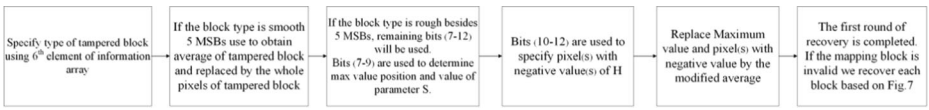

Fig. 3 The overview of watermark extraction and verification process

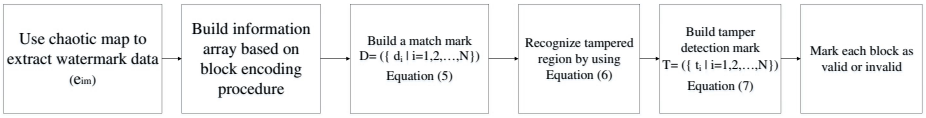

Fig. 4 The overview of image recovery process

Indeed, by applying above procedure, corresponding to each block I_i $(1 \le i \le N)$, the information array α_I with length six or twelve is built. Here, the aforementioned array called as watermarked data.

2.2 Key agreement

In order to enhance safety of the proposed scheme, the watermarked data ($\alpha_i i = 1, 2\ldots, N$) encrypted with a key that agreed between users. Indeed, a key agreement between users is an essential step in the proposed scheme. Users (i.e. user *A* and *B*, who are participating in watermarking process), apply an efficient and secure key agreement protocol by using their private key besides the partner public key. In this field, too many articles exist such as [1–4, 14, 18].

2.3 Watermark embedding

Assume that user *A* wants to embed watermark data ($\alpha_i i = 1, 2\ldots, N$), into image (*I*) and send it to user *B*. First, he/she build key agreed (*K*), then use it for encryption of watermarked data based on the following equation:

$$\widetilde{\alpha}_i = \alpha_i \oplus K, \quad i = 1, 2\ldots, N \tag{3}$$

Then, we use a two dimensional chaotic map, to determine exact position ($X(i)$, $Y(i)$) that encrypted information array $\widetilde{\alpha}_i$ will be embedded to it. The mathematical formula of this map is as below:

$$\begin{cases} X_i = mod\left(1-2\left(Y_{i-1}^2\right) \times 10^5, 1\right) \\ Y_i = mod\left(cos\left(6 \times arccos(X_{i-1}\right) \times 10^5, 1\right) \end{cases} \tag{4}$$

where mod is modulo operation that finds the reminder of one number by another. In order to determine initial value $\begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$, by using agreed key *K*, a number between zero and one will be generated, assume this number as X_0 (for this purpose, based on domain value of *K*, simple mathematical operator can be used). Then, obtain Y_0 by using below formula:

$$Y_0 = mod\left(\cos\left(6 \times \arccos(X_0) \times 10^5, 1\right)\right.$$

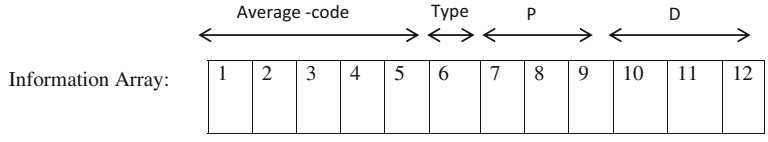

Fig. 5 Information Array (α_I)

Table 1 Blocks with the three *LSB* removed and their *CC*

Block Content [a_0, a_1, a_2, a_3]	Average	High Frequency (H)	*CC*		
			Average	Type	Detail
Proposed method					
11,12,12,13	12	−1,0,0,1	01100	0	
30,3,2,1	9	21,-6,-7,-8	01001	1	000110
11,12,13,14	**12**	**−1,0,1,2**	**01100**	**0**	
23,3,22,21	17	6,-14,5,4	10001	1	001000
Chen et al. method					
11,12,12,13	12	−1,0,0,1	01100	0	
30,3,2,1	9	21,-6,-7,-8	01001	1	001101
11,12,13,14	**12**	**−2,-1,0,1**	**01100**	**1**	**101001**
23,3,22,21	17	6,-14,5,4	10001	1	010010

We iterate Eq. (4) for N times, where N demonstrates the number of sub-blocks in the image and sequence $\begin{pmatrix} X_i \\ Y_i \end{pmatrix}$, $1 \le i \le N$ will be produced. Then, two sequence X_i and Y_i, $1 \le i \le N$ will be extracted. After that sort function used to arrange X_i and Y_i elements as an index of row and column that the information array of ith block is embedded in it, respectively. Next, the watermarked data is embedded into three least significant bits of each pixel in the mapping block. Note that, if the length of the information array is equal to 6, we use three LSB's of two pixels and otherwise we use three LSB's of four pixels.

2.4 Watermark extraction and verification

Watermark extraction is the reverse procedure of watermark embedding process. The receiver with the help of chaotic map (Eq. 4) and appropriate initial values can define exact position of blocks that watermarked data embedding is in it. The notation e_{im}, demonstrates the extracted watermarked data of the ith block, with size m. The same procedure in section 2.1 is done for received image and builds the watermarked data (information array) based on received image.

Table 2 Fulfill of positions 7 ~ 9

Status	S	Maximum Position	$P_{7\sim9}$
1	0	1	000
2	1	1	001
3	0	2	010
4	1	2	011
5	0	3	100
6	1	3	101
7	0	4	110
8	1	4	111

Table 3 Fulfill of positions 10 ~ 12

Status	The position(s) of pixel(s) that is (are) less than average	$D_{10\sim12}$
1	1	000
2	2	001
3	3	010
4	(1,2)	011
5	(1,3)	100
6	(2,3)	101
7	(1,2,3)	110

Note that all of the watermarked data, which extracted from the received images, may not be valid. Indeed, invalid watermarked data situations consist of modified pixels and pixels, which mapping blocks (that consist of their corresponding information) changed after transmission process. Hence, it cannot be used as appreciate tool for checking consistency of blocks. To get rid of this problem a match mark is defined ($D = \{d_i | i = 1, 2, \ldots, N\}$). For each block, d_i determined by comparing extracted watermark and the watermark which produced by the method in section 2.1.

$$d_i = \begin{cases} 0 & if\ e_{jm} = w_{im}\ \forall m \leq \vartheta_i \\ 1 & otherwise \end{cases} \tag{5}$$

where ϑ_i is the length of information array of the ith block and j, is the index of mapping block corresponding to the ith block. We use the following equation (Eq. 6) to better represent the location of tempering regions. If $t_i = 1$, the corresponding block is invalid, otherwise is valid.

$$t_i^0 = \begin{cases} 1 & if\ \left(d_i = 1\ ,\ \Gamma_i^D \geq \Gamma_j^D\right) \\ 0 & otherwise \end{cases} \tag{6}$$

Where j, is the index of mapping block corresponding to the ith block, the notations $\Gamma_i^D and\ \Gamma_j^D$ denote the number of valid blocks that are adjacent to ith and jth blocks in D respectively. The temper detection mark (*TDM*), $T = \{t_i | i = 1, 2, \ldots, N\}$ is,

$$t_i = \begin{cases} 0 & if\ t_i^0 = 1\ and\ \Gamma_i^{t_0} \leq 2 \\ 1 & if\ t_i^0 = 0\ and\ \Gamma_i^{t_0} \geq 5 \\ t_i^0 & otherwise \end{cases} \tag{7}$$

where $\Gamma_i^{t_0}$, denotes the number of valid blocks adjacent to ith block in the initial *TDM* t^0. After temper detection, all blocks mark as valid or invalid. In the following, we discuss image recovery procedure.

2.5 Image recovery

In this section, our efficient method for restoration of tampered image introduced. The proposed method contains two main stages. First, the recovery procedure of blocks with valid mapping blocks is done, which helps to have a better recovery of tampered blocks with invalid mapping blocks. Second, for the block recovery, which have invalid mapping blocks, the

information of adjacent pixels are used (Fig. 6). The detailed procedure of image recovery is as follows:

In the first stage of image recovery, our goal is restoration of tampered locations, which their mapping blocks are valid. After localization of these regions, following steps will be done.

Step 1: Based on the 6th element of extracted information array from corresponding mapping block, the tampered block can be marked as smooth or rough.
Step 2: If the block is smooth, five most significant bits (*MSBs*) of the extracted information array used to obtain the average of tampered block. Because, the value range of images is between 0 and 256, we fulfill positions 6 to 8 with zero to meet this requirement.
Step 3: The whole pixels value in the tampered blocks will be replaced by the average obtained in previous step.
Step 4: If the block is rough, besides extraction of five *MSBs*, the remaining bits (7–12) will be used. The extracted bits from positions 7–9 are used to determine maximum pixel value position and value of parameter *S*, which shown the distance between maximum pixel value and average of pixel(s) with negative value(s) of *H*.
Step 5: The bits at positions 10–12 are used to specify position(s) of pixel(s) with negative value(s) of *H* relative to maximum pixel value position. The position(s) could be determined with decoding the bits stored in these positions (see Fig. 7 and Table 3).

Remark: One of the innovations in this article is the use of special image block features in definition of information array. The information about position of maximum pixel value, the distance between maximum pixel value and average of pixel(s) with negative value(s) of *H* besides the exact position(s) of aforementioned pixel(s) relative to maximum pixel value position are stored in information array. Then, this information embeds in the mapping positions of the corresponding block. The advantage of storing this information is that we can achieve good image recovery with help of these features, because instead of replacing all pixel values in the block with average value, we put nearest possible value based on the aforementioned feature's state.

Step 6: Based on the bit values in position 7–12, we can obtain *S*, maximum position, and position(s) of pixel(s) with negative amount of *H*. If the value of *S* equal to zero, the maximum value and pixel(s) with negative value(s) of *H* will be replaced by the modified average as follows, otherwise the pixel values in the tempered block will be replaced by the extracted average that expanded to 8 bits by adding zero to the bit positions 6–8. The two most

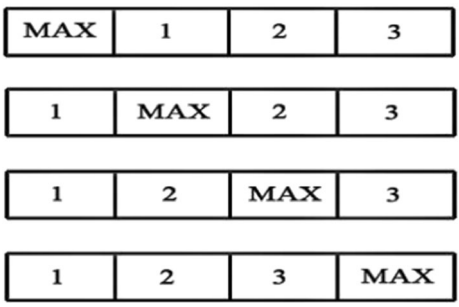

Fig. 6 Position(s) of negative element(s) relative to maximum

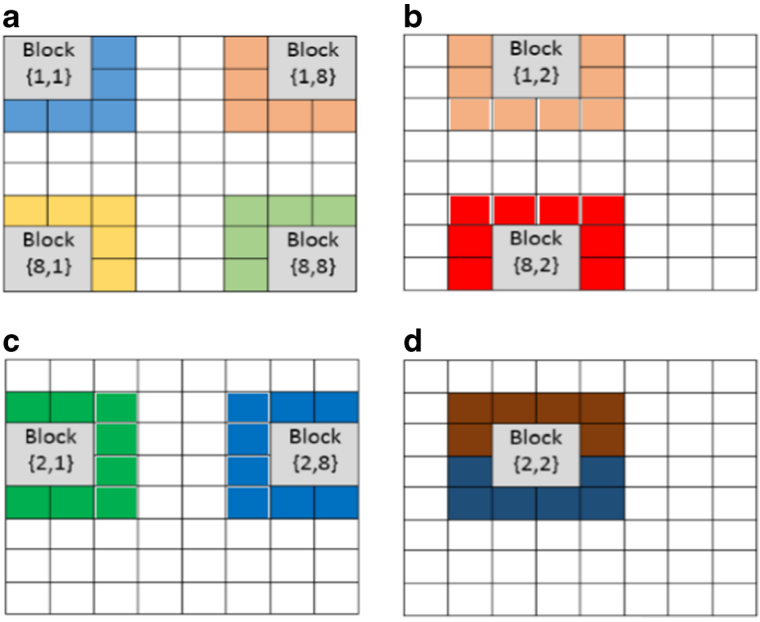

Fig. 7 Conditions of block recovery. **a** for blocks that placed in a corner, the average of five adjacent pixels are used, **b** for blocks placed in first or last row, the average of four adjacent pixels (2 east, 2 west) are used, **c** for blocks in first or last column, the average of four pixels (2 south 2 north) are calculated, d: for inner blocks, the average of all twelve adjacent pixels are measured

significant bit of average will be replaced by "11" for maximum value pixel and "00" for pixel(s) with negative value(s) of H, respectively. Similar to procedure done in step 2, bit positions 6–8 are fulfilled with zero. The remaining pixel(s) in the block will be replaced by the extracted average from information array that its 6–8 bit positions filled with zero.

After completing over mentioned steps, the first round of recovery phase will be completed, and in the second round we deal with the conditions that the mapping block of specific block is not valid. In this case we use pixels of valid blocks that are adjacent to the tempered blocks (Fig. 6).

Step 7: Based on positions of tampered blocks, the number of adjacent pixel will be changed as shown in Fig. 6. After specifying the number of adjacent pixels, validation of their corresponding blocks will be checked to specify the target pixels set.
Step 8: We use result of previous step to determine which pixels participate in calculating the average and the calculated value replaced by previous amount of pixels. Finally, the reconstructed image can be obtained with acceptable quality (Table 6).

3 Experimental results

In order to prove the effectiveness of proposed method, some experiments and comparisons are done with latest researches in [8, 12, 27]. The proposed method is checked by two standard image formats BMP and Tiff. Although, the experimental results are based on Tiff format. Several measurements such as, code-length (*bpp*: bit per pixel) and code-quality [29] (*PSNR* between

Table 4 Comparison of coding efficiency for different images

Image	Watermark Payload (bpp)					Watermark Image Quality						
	Proposed	[8]	[27]	[12]	[5]	Proposed	[8]	[27]	[12]	[23]	[5]	[25]
Boat	1.59	1.62	3	1.13	2	43.99	44.20	37.89	48.59	37.56	44.65	36.51
Lena	1.57	1.63	3	1.24	2	37.00	44.12	37.92	48.00	34.76	45.75	35.43
Peppers	1.57	1.62	3	1.25	2	44.94	43.81	37.92	48.01	39.12	43.24	37.12
Barbara	1.73	1.93	3	1.37	2	44.28	41.73	37.92	47.10	38.49	44.20	34.97
Baboon	1.71	2.19	3	1.59	2	37.19	39.79	37.92	45.81	36.73	37.29	36.94

recovered image and original one) are used to prove coding efficiency. Moreover, similar metrics [30] (*PSNR* between watermarked image and original one) and watermark payload (*bpp*), are applied to show invisibility property of watermarked image. After that, in section 3.2, some visual results are presented, which refer to temper detection and recovery performance. Then, for proving efficiency of proposed scheme, the chi-square test is used. Finally, the quality of index is calculated to prove the similarity between reconstructed image and original one.

3.1 Code efficiency and invisibility

As mentioned before, we do this test to prove invisibility and efficiency of proposed method. We use bit per pixel (*bpp*) and *PSNR* in order to prove over mentioned goals. As described in section 2, there is relation between code-length and image type, the smoother the image get, the smaller value shown by code-length. By defining accurate metric for distinguishing rough and smooth blocks, the watermark payload of our scheme becomes minimum. This fact is shown in Table 4 and implies optimal bandwidth consumption. Although, applying different code-length proposed in [8, 15], but incorrect distinguishing of smooth blocks as rough blocks that is result of inefficient metrics, causes much bandwidth consumption (Table 1). Our code quality is almost the best alongside others latest proposed schemes. Indeed, we portioned image into 2×2 blocks, which increases precision of temper detection and leads to better recovery of tempered image. The results of invisibility test with comparison of experimental results provided in Table 5. Furthermore, Comparison of image recovery phase between proposed and Chen et al. method provided in Table 6.

Table 5 Performance comparison of invisibility for different images

Image	Code-Length (bpp)				Code-Quality					
	Proposed	[8]	[27]	[12]	Proposed	[8]	[27]	[12]	[5]	[25]
Boat	1.59	1.62	2.6	0.82	36.69	35.29	30.51	33.41	35.59	37.83
Lena	1.57	1.63	2.6	0.93	35.79	33.04	32.43	33.96	36.76	36.02
Peppers	1.57	1.62	2.6	0.94	36.23	32.21	30.72	32.74	33.87	36.74
Barbara	1.73	1.93	2.6	1.06	36.31	29.97	28.18	26.57	35.98	38.78
Baboon	1.71	2.19	2.6	1.28	35.65	27.48	25.71	25.56	34.12	36.97

Table 6 Comparison of image recovery phase between proposed and Chen et al. method

Block Content [a_0, a_1, a_2, a_3]	Average	Tampered value(s)	Recovery Block [a_0, a_1, a_2, a_3]
Proposed method			
11,12,12,13	12	25,12,12,3	12,12,12,12
30,3,2,1	**9**	**15,23,25,26**	**25,1,1,1**
11,25,1,2	10	1,15,23,26	10,26,1,1
25,3,4,1	8	1,29,4,2	24,0,0,0
Chen et al. method [8]			
11,12,12,13	12	−1,0,0,1	12,12,12,12
30,3,2,1	**9**	**21,-6,-7,-8**	**9,9,9,9**
11,25,1,2	10	−2,-1,0,1	12,12,12,12
25,3,4,1	8	6,-14,5,4	8,8,8,8

3.2 Tamper detection and recovery

In order to show the robustness of tamper detection and recover quality of proposed scheme, we use Lena, Barbara, Baboon with size of 512 × 512 for this test. The watermarked Lena, Barbara and Baboon generated by the proposed scheme are shown in Fig. 8(a)–(c). Three tampered images shown in Fig. 8(d)–(f). As the results in Fig. 9 demonstrate, our proposed scheme has a good performance in tamper localization and recovery.

In Fig. 9, we have provided the tampered localization and image reconstruction results. As it's clear in this picture, the proposed scheme has a convincing result.

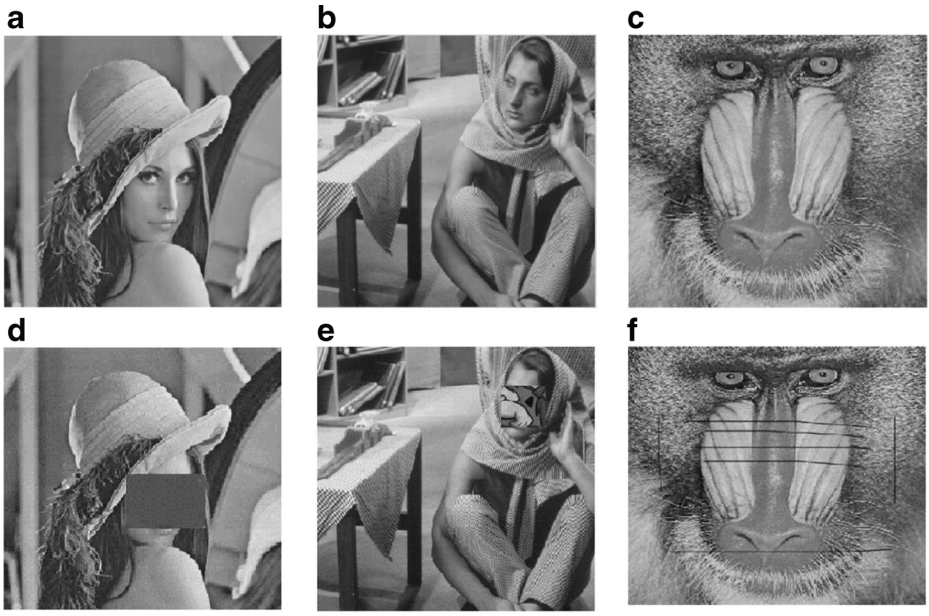

Fig. 8 Watermarked and tempered images. **a** Watermarked Lena, **b** Watermarked Barbara, **c** Watermarked Baboon, **d** Tempered Lena, **e** Tempered Barbara, **f** Tempered Baboon

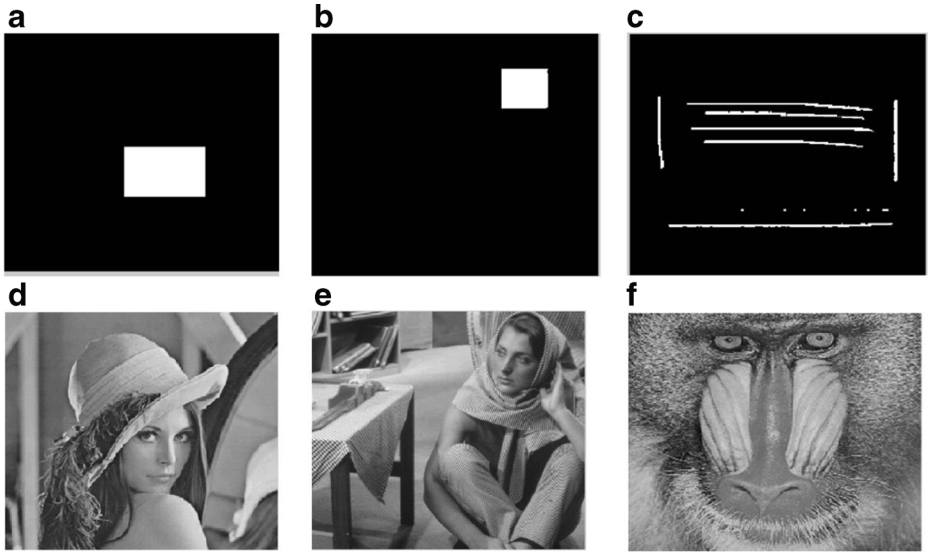

Fig. 9 Tampered detection and recovery images. **a** Tampered Lena, **b** Tampered Barbara, **c** Tampered Baboon, **d** Recovered Lena, **e** Recovered Barbara, **f** Recovered Baboon

Finally, In Fig. 10 we provide a comparison between our proposed method and some recent schemes.

3.3 Chi-square test

In the proposed method, we use *LSBs* of each block (6 ~ 12 bits) to embed watermark data array, therefore, the *LSB* plan of image would be changed after embedding process. Chi-square test, [19] as a statistical analysis, helps us to understand whether a difference between expected signal and observed signal exist or not. The result of this test is shown in Fig. 11. If the red line come close to 1, it means that, there is a data, which is embedded in host image, in contrast, if it's close to zero, it means existence of hidden secret data in the host image can't observe [9].

3.4 Quality index

In order to measure, quality of recovered watermark image, we use quality index [19] that is calculated based on below formula:

$$Q = \frac{4\sigma_{HT}\, H' T'}{(\sigma_H^2 + \sigma_T^2)\left[H'^2 + T'^2\right]} \tag{8}$$

Where,

$$H' = \frac{1}{N}\sum_{i=1}^{N} H_i,\; T' = \frac{1}{N}\sum_{i=1}^{N} T_i \tag{9}$$

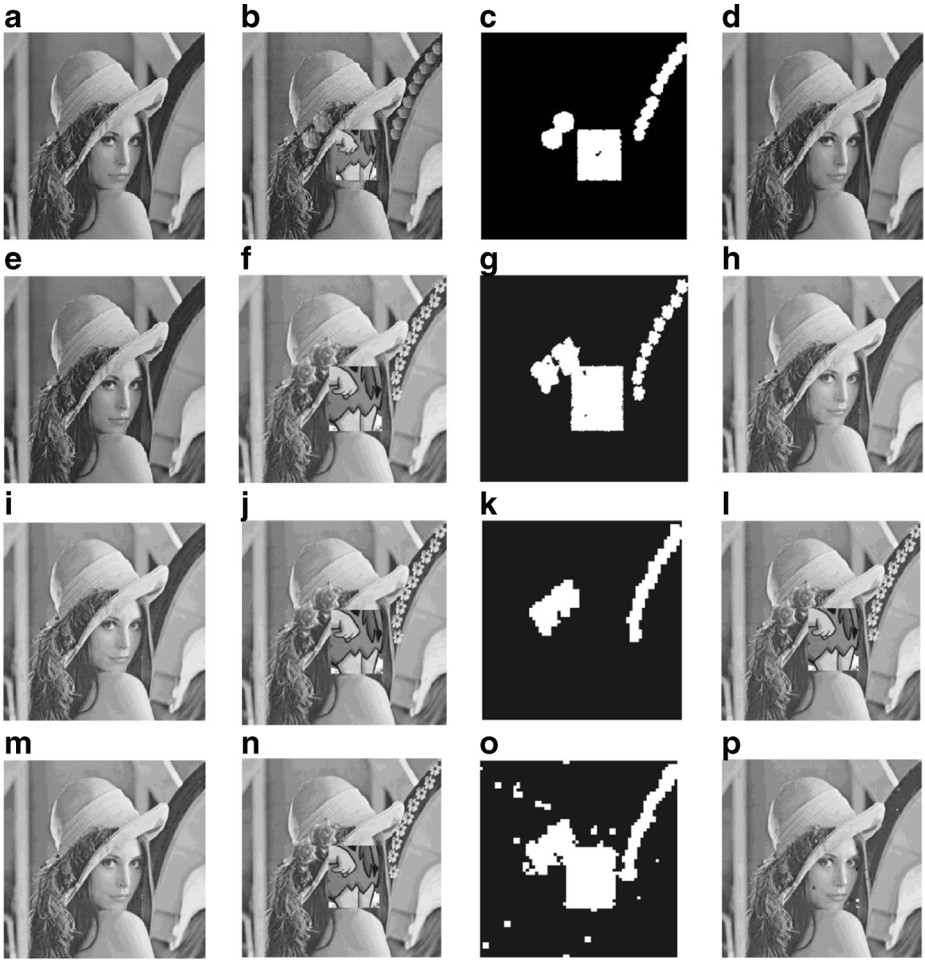

Fig. 10 Comparison of proposed method with schemes in [8, 12, 27]. (a ~ d): Lena original image, Lena tampered image, Lena tamper detection, Lena recovery image. (e ~ h): Lena original image, Lena tampered image, Lena tamper detection based [8], Lena recovery image based [8]. (i ~ l): Lena original image, Lena tampered image, Lena tamper detection based [27], Lena recovery image based [27]. (m ~ p): Lena original image, Lena tampered image, Lena tamper detection based [12], Lena recovery image based [12]

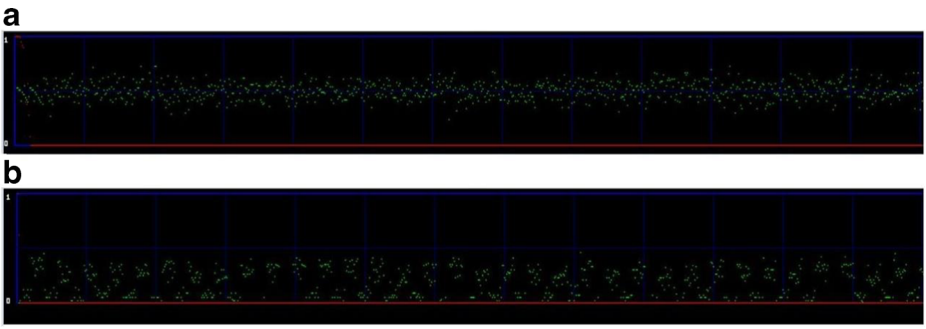

Fig. 11 Chi-square test **a** original Lena image, **b** Watermarked Lena image

Table 7 Quality of index results for standard images

Images	Quality of index Value
Baboon	0.9920
Barbara	0.9979
Boat	0.9935
Lena	0.9989
Pepper	0.9971

$$\sigma_H^2 = \frac{1}{N-1}\sum_{i=1}^{N}(H_i - H')^2 \tag{10}$$

$$\sigma_T^2 = \frac{1}{N-1}\sum_{i=1}^{N}(T_i - T')^2 \tag{11}$$

In the above equation, n is the number of pixels in the image, H is the host image and T is the recovered watermark image. The value of Q is between 1 and −1. If the calculated value be −1, it means that, the host image and the recovered watermark image are not similar, but if it equals to 1, it means that two images are identical. The calculated values of this metric are shown in Table 7.

4 Conclusion

We have presented a novel fragile watermarking scheme based on chaotic maps that generates the embedded data with as few bits as possible while maintains the standards of good watermarking scheme such as accurate localization of tampered regions, satisfying recovery quality (both of them shown in Fig. 9), and optimal watermark payload (Table 1). The host image is portioned into sub-blocks with fix size of 2 × 2 pixels to improve the accuracy of proposed method in detection of tampered region. The chaotic maps are used in block mapping phase to enhance security of the proposed scheme. A novel efficient metric for determining rough or smooth blocks are proposed, which has led to at least three advantageous. First, "decreasing of bandwidth consumption" due to minimum watermark payload length. Indeed, the length of embedded data varies based on characteristic of block (6 bits for smooth blocks and 12 bits for rough blocks will be embedded in a host image). Some more advantageous are, "better tamper detection" and "self-recovery with high quality", because of special image features which used in definition of information array. Our method, provides basic of watermarking scheme such as, invisibility, recover quality and security. Experimental results have proved that our method is powerful in tamper detection, self-recovery and robust against the known watermarking attacks.

References

1. Arshad H, Nikooghadam M (2014) Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems. J Med Syst 38(12):1–12

2. Arshad H, Nikooghadam M (2015) Security analysis and improvement of two authentication and key agreement schemes for session initiation protocol. J Supercomput 71(8):3163–3180
3. Arshad H, Nikooghadam M (2016) An efficient and secure authentication and key agreement scheme for session initiation protocol using ECC. Multimedia Tools Appl 75(1):181–197
4. Arshad H, Teymoori V, Nikooghadam M, Abbassi H (2015) On the security of a two-factor authentication and key agreement scheme for telecare medicine information systems. J Med Syst 39(8):1–10
5. Barani MJ, Ayubi P, Jalili F, Valandar MY, Azariyun E (2015) Image forgery detection in contourlet transform domain based on new chaotic cellular automata. Secur Commun Netw 8(18):4343–4361
6. Chen B, Shu H, Coatrieux G, Chen G, Sun X, Coatrieux JL (2015) Color image analysis by quaternion-type moments. J Math Imaging Vis 51(1):124–144
7. Chen F, et al. (2012) Self-recovery fragile watermarking scheme with variable watermark payload, in Digital Forensics and Watermarking. Springer, p 142–155
8. Chen F et al (2014) Chaos-based self-embedding fragile watermarking with flexible watermark payload. Multimedia Tools Appl 72(1):41–56
9. El-Loco G: A few tools to discover hidden data. http://www.guillermito2.net/stegano/tools/index.html
10. Fridrich J and Goljan M (1999) Protection of digital images using self embedding. In: Symposium on Content Security and Data Hiding in Digital Media. Newark, NJ, USA
11. Gong L, Liu X, Zheng F, Zhou N (2013) Flexible multiple-image encryption algorithm based on log-polar transform and double random phase encoding technique. J Mod Opt 60(13):1074–1082
12. Huo Y, He H, Chen F (2012) Alterable-capacity fragile watermarking scheme with restoration capability. Opt Commun 285(7):1759–1766
13. Hwang HE (2012) Optical color image encryption based on the wavelength multiplexing using cascaded phase-only masks in Fresnel transform domain. Opt Commun 285(5):567–573
14. Kumari S, Li X, Wu F, Das AK, Arshad H, Khan MK (2016) A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps. Futur Gener Comput Syst 63:56–75
15. Lee T-Y, Lin SD (2008) Dual watermark for image tamper detection and recovery. Pattern Recogn 41(11): 3497–3506
16. Liao X, Shu C (2015) Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels. J Vis Commun Image Represent 28:21–27
17. Mollaeefar M, Sharif A, Nazari M (2015) A novel encryption scheme for colored image based on high level chaotic maps. Multimedia Tools Appl 1–23
18. Nikooghadam M, Jahantigh R, Arshad H (2016) A lightweight authentication and key agreement protocol preserving user anonymity. Multimedia Tools Appl
19. Provos N and Honeyman P (2001) Detecting steganographic content on the internet. Center for Information Technology Integration
20. Qian Z et al (2011) Image self-embedding with high-quality restoration capability. Digit Signal Process 21(2):278–286
21. Qin C, Chang C-C, Chen P-Y (2012) Self-embedding fragile watermarking with restoration capability based on adaptive bit allocation mechanism. Signal Process 92(4):1137–1150
22. Qin C, Chang CC, Chen YC (2013) Efficient reversible data hiding for VQ-compressed images based on index mapping mechanism. Signal Process 93(9):2687–2695
23. Saha C, Hossain MF, Abdullah BS (2016) Improving the security of spatial domain based digital image watermarking using chaotic map and cellular automation. Int J Image Graph Signal Process 8(1):51
24. Sharif A, Mollaeefar M, Nazari M (2016) A novel method for digital image steganography based on a new three-dimensional chaotic map. Multimedia Tools Appl 1–19
25. Singh P and Agarwal S (2016) A self recoverable dual watermarking scheme for copyright protection and integrity verification. Multimedia Tools Appl 1–40
26. Yang C-W, Shen J-J (2010) Recover the tampered image based on VQ indexing. Signal Process 90(1):331–343
27. Zhang X et al (2011) Reference sharing mechanism for watermark self-embedding. IEEE Trans Image Process 20(2):485–495
28. Zhou N, Li H, Wang D, Pan S, Zhou Z (2015) Image compression and encryption scheme based on 2D compressive sensing and fractional Mellin transform. Opt Commun 343:10–21
29. Zhou N, Wang Y, Gong L (2011) Novel optical image encryption scheme based on fractional Mellin transform. Opt Commun 284(13):3234–3242
30. Zhou N, Zhang A, Zheng F, Gong L (2014) Novel image compression–encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing. Opt Laser Technol 62:152–160

Mahboubeh Nazari received B.Sc. degree from Ferdowsi University of Mashhad, Iran, in 2006, M.Sc. degree from Ferdowsi University of Mashhad, Iran, in 2008 and Ph.D. degree from Ferdowsi University of Mashhad, Iran, in 2013. She is Adjunct professor in Department of Mathematics at Ferdowsi University of Mashhad, Iran. Her research focuses on Dynamical Systems, Chaos theory and it's applications in Cryptography, Network Security and data Security.

Sharif Amir was born in 1990 in Birjand, south Khorasan Province. He received his B.Sc. degree in information technology from Birjand University in June 2012. Nowadays, he study secure communication in Imam Reza International University of Mashhad, attendant for M.Sc. degree. His filed of interest are image processing, steganography, cryptography, Covert channel and security.

Mollaeefar Majid was born in 1989 in Gonbad-e Kavus, Golestan Province. He received his B.Sc. degree in information technology from Tabari University in June 2013. Nowadays, he study secure communication in Imam Reza International University of Mashhad, attendant for M.Sc. degree. His filed of interest are image processing, cloud security, steganography, Covert channel and security.