Multimed Tools Appl (2014) 72:41–56
DOI 10.1007/s11042-012-1332-5

Chaos-based self-embedding fragile watermarking with flexible watermark payload

Fan Chen · Hongjie He · Heng-Ming Tai · Hongxia Wang

Published online: 25 December 2012

Abstract This paper proposes a self-embedding watermarking scheme that reduces the watermark payload significantly while maintaining good recovery quality and security. The embedded watermark contributes to the tamper detection and content recovery and is composed of only the compression codes of the image content. The compression codes with variable length are generated according to the roughness of the image. To improve the security, a chaos-based pseudorandom sequence generator is adopted to generate block-mapping sequence and encrypt compression codes. The proposed method takes into account the invisibility, recovery quality, and security using the flexible watermark payload, which preserves sufficient information of the image block with as few bits as possible. Experimental results demonstrate that the proposed scheme not only outperforms conventional self-embedding fragile watermarking algorithms in tamper detection and recovery, but also improve the security against the various counterfeiting attacks.

Keywords Fragile watermarking · Self-embedding · Flexible watermark payload · Chaos

1 Introduction

Self-embedding fragile watermarking is designed to achieve digital content authentication and to recover the original content in the tampered regions by imperceptibly embedding additional data into the host image [3]. It usually divides a host image into blocks of the same size and generates the recovery data of a block by compressing the block content. In this paper, the recovery data are called compression code (CC). Examples include the quantized DCT coefficients [3], VQ indexing [11], and average intensity [5, 7, 13]. The number of CC bits generated by these methods is fixed for all blocks. The fixed length

F. Chen · H. He (✉) · H. Wang
School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China
e-mail: hehojie@126.com

H.-M. Tai
Department of Electrical Engineering, University of Tulsa, Tulsa, OK 74104, USA

constraint results in the drawback that the CC is overmuch for smooth blocks, but is inadequate for a rough block [9]. The overmuch code often increases the watermark payload (WP), and the inadequate code impairs the quality of reconstructed image. To address this problem, the multi-level encoding was proposed to generate the CC with variable length for various types of blocks [9, 10]. However, since the watermark embedding capacity was fixed [9, 10], the amount of the CC bits was still fixed for different images. These schemes [9, 10] must adjust the block classification to guarantee that the numbers of the CC are exactly suitable for the fixed watermark payload. Moreover, these schemes still lead to the situation that there are some blocks of which CC are overmuch in the smooth image and inadequate in the rough image.

To address the problems above, authors proposed the DCT-based alterable-capacity self-recovery fragile watermarking schemes [2, 6]. The blocks of size 8×8 pixels were classified into eight types according to the roughness of the blocks in [6]. The CC of a block included three parts: 20-bit significant-code, 3-bit type-code and detail-code with variable number of bits ranging from 0 to 78. The alterable-length watermark was divided into three parts and embedded in other three blocks. To improve the quality of recovered images, two copies of the significant-code of each block were embedded in different blocks, and the image inpainting method was adopted to recover the tampered blocks whose two copies of significant-code embedded in other blocks had been destroyed. However, since the size of block is 8×8 pixels, the accuracy of tamper localization would be impaired, and the quality of the recovered image would be degraded when the ratio of tampered regions becomes larger.

In most mentioned self-embedding schemes, the *WP* is more than the average length of the CC [2, 5–7, 9–11, 13] due to the fact that some redundant information is introduced. According to the different role of its plays, the redundant information can be divided into three categories. (1) To improve security, Ref. [13] added 2-bit key-based data for each 2×2 block to improve the ability against the average-attack proposed by Chang et al. [1]; (2) To resolve the tampering coincidence problem [13], more than one copy of the CC (part or all) is embedded in the host image. For example, Yang [11] and Lee [7] were embedded four and two copies of watermark of the whole image respectively, Huo [6] embedded the significant-code of each block twice, and Zhang [9, 13] adopted the reference sharing mechanism. These strategies improve the quality of the recovered image especially when the tampered area is larger; (3) To resolve the tamper detection problem, the authentication data are added for each block [7, 9–11, 13]. Lin [7] added the 2 bits authentication data for each block of size 2×2 pixels, and Yang [11] and Zhang et al. [9, 13] added 64 and 32 bits authentication data for each block of size 8×8 pixels, respectively. A common feature of them embedded the authentication data of each block in the same block, so these methods [7, 9, 11, 13] are vulnerable to the collage attack proposed by Fridrich [4]. That is, these schemes fail to detect and recover the collaged regions in the test image. Also, the tampered regions besides the collaged ones would not be recovered by the self-embedding schemes adopting the reference sharing mechanism [9, 13]. To resist the collage attack and decrease WP, Qin [10] produced the authentication data of each block using the image hashing method with a folding operation, and judged the integrity of the blocks by a voting-based strategy, which was proposed by Zhang [12]. This method can identify the blocks containing fake contents as long as the tampered area is not extensive [12]. However, the detection performance degrades with the increase of the tamper ratio. By adding the redundant information, although the security, detection ability and recovery quality could be improved, the quality of watermarked image might be degraded.

This paper proposes a self-embedding scheme with flexible WP to reduce the amount of embedded data while preserving sufficient information of the content. The original image is

divided into blocks of size 2×2 pixels to improve the accuracy of localization. The 2×2 blocks are classified into the smooth block and the rough one, and the CC of each block is allocated the proper number of bits according to its need. To further improve the security, a pseudorandom sequence generated by the chaotic mapping is used to encrypt the CC, and obtain the block-mapping sequence. The proposed scheme achieves the flexible WP and does not need the complicated block-classification adjusting algorithm. As a result, the WP of the proposed scheme equals to the average length of the CC, and the recovery quality and security are improved due to the fact that all the embedded data contribute to the content recovery and tamper detection.

2 Proposed method

The proposed scheme adopts the block-neighborhood tamper detection and recovery strategies proposed in [5]. The self-embedding procedure focusing on the block encoding, watermark embedding, watermark extraction and verification is described below.

2.1 Block encoding

This work partitions the original image $\boldsymbol{X}$ into N non-overlapping 2×2 blocks X_i (i=1,2,…, N) to improve the localization accuracy. Let B denote the content (5 most significant bit (MSB) planes) of a block of 2×2 pixels. The CC of a block content includes average-code, type-code, and detail-code, as shown in Fig. 1. The length of CC is not more than 12 bits and not less than 6 bits. The average-code and type-code are required for all block, but the detail-code may be omitted for a smooth block because most of the coefficients in its high-frequency component are zeros. The procedure generated the CC of block content B is represented as,

$$(C,\ v) = P_{code}(B) \tag{1}$$

where C is the CC of block content B, v is the code length, and $P_{Code}()$ denotes the proposed compression encoding process, which consists of three steps.

Step 1: Average-code. Let $B = \begin{pmatrix} b_0 & b_1 \\ b_2 & b_3 \end{pmatrix}$, b_k (k=0,1,2,3) is an integer ranges from 0 to 31. The average value of the block content is,

$$a = round\left(\frac{1}{4}\sum_{k=0}^{3} b_k\right) \tag{2}$$

where the *round* (.) returns the nearest integer of the argument. Since the average value must fall into [0, 31], the first five bits of the CC are computed as

$$c_m = mod\left(\left\lfloor a/2^{m-1}\right\rfloor, 2\right), \quad m = 1, \cdots, 5 \tag{3}$$

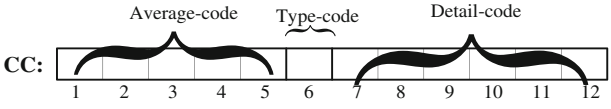

Fig. 1 Composition of compression code

where *mod*(,) is the modulo operation, and $\lfloor x \rfloor$ is the largest integer less than or equal to x.

Step 2: Type-code. The high-frequency component of block content is generated by,

$$H = \begin{pmatrix} h_0 & h_1 \\ h_2 & h_3 \end{pmatrix} = \begin{pmatrix} b_0 - a & b_1 - a \\ b_2 - a & b_3 - a \end{pmatrix} \tag{4}$$

Let h_{k1} and h_{k2} be the two largest in H, the difference between the two largest values and the two smallest values in H is,

$$dif = (h_{k1} + h_{k2}) - \left(\sum_{k=0}^{3} h_k - (h_{k1} + h_{k2}) \right) \tag{5}$$

If dif<5, the corresponding block is considered as smooth; otherwise, it is considered as rough. The sixth bit in the CC denotes the type of the block. That is, if the block is smooth, c_6=0 and v=6, otherwise c_6=1 and v=12.

Step 3: Detail-code. The detail-code (i.e., $c_{7\sim12}$) is generated only for those rough blocks. The $c_{7\sim9}$ is used to store the position of the two largest values,

$$c_{7\sim9} = \begin{cases} 110 & , \quad if\,(k_1 + k_2 = 3)\&(k_1 k_2 = 0) \\ [\![k_1 + k_2]\!]_2 & , \quad otherwise \end{cases} \tag{6}$$

And the $c_{10\sim12}$ is used to store a 3-bit uniform-quantized value of the sum of the two largest values in H. That is,

$$c_{10\sim12} = \left[\!\left[\left\lfloor \left| \frac{h_{k1} + h_{k2}}{4} \right| \right\rfloor \right]\!\right]_2 \tag{7}$$

Table 1 shows the typical coding expression for certain 2×2 blocks where the 3 LSB has been removed. The average intensity, high-frequency and CC are also shown.

2.2 Watermark embedding

The proposed watermark embedding process is illustrated in Fig. 2. The details are described as follows.

Table 1 Blocks with the three LSB removed and their CC

Block Content $[b_0,b_1,b_2,b_3]$	Average	High-frequency $B=[h_0,h_1,h_2,h_3]$	CC		
			Average	Type	Detail
11,12,12,13	12	−1,0,0,1	01100	0	
30,3,2,1	9	21,−6,−7,−8	01001	1	001101
23,3,22,21	17	6,−14,5,4	10001	1	010010
15,3,4,16	10	5,−7,−6,6	01010	1	110011
7,15,12,10	11	−4,4,1−1	01011	1	011001
3, 20,4,21	12	−9,8,−8,9	01100	1	100100
4,3,20,21	12	−8,−9,8,9	01100	1	101100

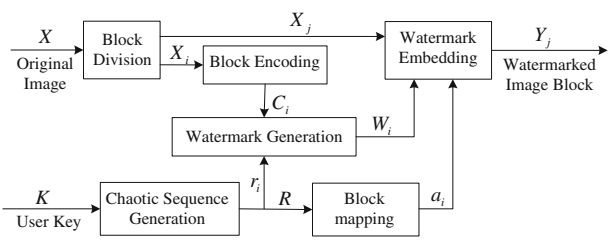

Fig. 2 Watermark embedding process

(1) *Block Division*. The original image X of size 2 m×2n pixels is partitioned into N non-overlapping 2×2 blocks $X = \{X_i | i - 1, 2, \ldots, N\}$.

(2) *Chaotic sequence generation*. According to the user key K, the pseudorandom sequence $R=\{r_i|i=1,2,\ldots,N\}$ is generated using the chaos-based pseudorandom sequence (CPRS) generator defined in [8].

$$R = mod\big(F(z,p) + F(z',p'), 2^{32}\big) \tag{8}$$

where $F(z, p)$ is the discrete piecewise linear chaotic maps, which is defined as,

$$Z_{k+1} = \begin{cases} \left\lfloor \frac{2^{32}\times z_k}{p} \right\rfloor, & 0 \le z_k < p \\ \left\lfloor \frac{2^{32}\times (z_k - p)}{2^{31} - p} \right\rfloor, & p \le z_k < 2^{31} \\ \left\lfloor \frac{2^{32}\times \left(2^{32} - z_k - p\right)}{2^{31} - p} \right\rfloor, & 2^{31} \le z_k < 2^{32} - p \\ \left\lfloor \frac{2^{32}\times \left(2^{32} - z_k\right)}{p} \right\rfloor, & 2^{32} - p \le z_k \le 2^{32} \end{cases} \tag{9}$$

where z_k denotes the discrete state ranging from 0 to 2^{32}–1, and $0<p<2^{31}$ is the discrete control parameter. That is, the seed of the CPRS generator is composed of two initial-values (z, z') and two control parameters (p, p'). Therefore, the secret key K is a binary stream with the length of 126 (2×32+2×31) bits.

(3) *Block encoding*. According to (1) described in Section 2.1, the CC of each block X_i is generated,

$$(C_i, v_i) = P_{code}(\lfloor X_i/8 \rfloor) \tag{10}$$

(4) *Watermark generation*. For each block X_i, the CC is encrypted to generate the watermark $W_i=(w_{im}|m=1,2,\ldots,v_i)$,

$$w_{im} = c_{im} \oplus r_{im}, \quad m = 1, \ldots, v_i \tag{11}$$

where r_{im} is computed by the value of r_i in the pseudorandom sequence R obtained by (8),

$$r_{im} = mod(z_i/2^m, 2), m = 1, \ldots, v_i \tag{12}$$

(5) *Block mapping*. By sorting out the the pseudorandom sequence R, an ordered index sequence A such that $r_{a1}\le r_{a2}\ldots\le r_{aN\text{-}1}\le r_{aN}$ is obtained,

$$A = \{a_i | i = 1, 2, \ldots, N\} \tag{13}$$

where $a_i \in [1, N]$, and $ai \neq ai$, for $\forall\ i \neq i'$. The block mapping (X_i, X_j) (i, $j \in [1, N]$) is generated by assigning the index of the block X_j be $j=a_i$.

(6) *Watermark embedding*. The watermark W_i of block X_i are embedded in its mapping block X_j ($j=ai$). To make the WP of a block flexible, if its length is 12 bits, the watermark W_i are embedded in the 3 least significant bit (LSB) planes of X_j; otherwise, it is inserted in the first LSB planes and the part of the second LSB planes of X_j. The watermarked block $Y_j=\{y_{jk}|k=0,1,2,3\}$ is obtained by one of the following two cases. If $v_i=12$,

$$y_{jk} = 8\lfloor x_{jk}/8 \rfloor + 4w_{i(k+9)} + 2w_{i(k+5)} + w_{i(k+1)}, k = 0, 1, 2, 3 \tag{14}$$

Otherwise,

$$y_{jk} = \begin{cases} \lfloor x_{jk}/4 \rfloor \times 4 + 2w_{i(k+5)} + w_{i(k+1)} & , \quad k = 0, 1 \\ \lfloor x_{jk}/2 \rfloor \times 2 + w_{i(k+1)} & , \quad k = 2, 3 \end{cases} \tag{15}$$

2.3 Watermark extraction and verification

The watermark extraction is the reverse process of watermark embedding. The extracted watermark $E_i=(e_{im})$ from the tested image Y_i can be obtained by,

$$e_{im} = \begin{cases} mod\left(y_{i(m-1)}, 2\right) & , \quad m = 1, 2, 3, 4 \\ mod\left(\left\lfloor y_{i(m-5)}/2 \right\rfloor, 2\right) & , \quad m = 5, 6, 7, 8 \\ mod\left(\left\lfloor y_{i(m-9)}/4 \right\rfloor, 2\right) & , \quad m = 9, 10, 11, 12 \end{cases} \tag{16}$$

Note that all the bits in the extracted watermark E_i of (16) are not always valid. If the invalid watermark data extracted by (16) are used to judge the consistency of a block, the valid blocks may be wrongly considered as the mismatch with high probability. This often leads to the poor performance of tamper detection. Therefore, the match mark $D=\{d_i|i=1, 2, \ldots,N\}$ is constructed only by the valid watermark data extracted by (16). For each block Y_i, the d_i is determined by comparing the watermark W_i computed by (11) with the extracted watermark E_j,

$$d_i = \begin{cases} 0 & , \quad if\ w_{im} = e_{jm} \forall m \le v_i \\ 1 & , \quad otherwise. \end{cases} \tag{17}$$

where, v_i is the length of the CC of block Y_i obtained by (10). The tamper detection mark (TDM) $T=\{t_i|i=1,2,\ldots,N\}$ is used to represent the location of tampering. If $t_i=1$, the corresponding test block Y_i is invalid, otherwise it is valid. Adopted by our previous work of [5], the initial TDM $T^0 = \left(t_i^0 | i = 1, 2, \ldots, N\right)$ is assigned according to the match mark D,

$$t_i^0 = \begin{cases} 1 & , \quad if\left(d_i = 1\right) \& \left(\Gamma_i^D \ge \Gamma_i^D\right) \\ 0 & , \quad otherwise \end{cases} \tag{18}$$

where $j=a_i$, Γ_j^D and Γ_j^D denote the number of nonzero pixels that are adjacent to the i^{th} and j^{th} pixel in the D respectively. The TDM $T=\{t_i|i=1,2,\ldots,N\}$ is,

$$t_i = \begin{cases} 0 & , \quad if\left(t_i^0 = 1\right) \& \left(\Gamma_i^{T^0} < 2\right) \\ 1 & , \quad if\left(t_i^0 = 0\right) \& \left(\Gamma_i^{T^0} < 2\right) \\ t_i^0 & , \quad otherwise \end{cases} \tag{19}$$

where $\Gamma_i^{T^0}$ denotes the number of nonzero pixels that are adjacent to the i^{th} pixel in the initial TDM T^0.

After tamper detection, all blocks in test image are marked as either valid or invalid. The recovery procedure is only for the invalid blocks. The invalid block X_i is recovered by the CC from its mapping block if the mapping block of X_i is valid, otherwise it is recovered by the average intensity of the neighboring valid pixels of block X_i. Details of the recovery procedure are described in [5].

3 Experimental results

Extensive experiments were conducted to demonstrate the effectiveness of the proposed scheme and compare with the latest schemes [13] and [6] in the performance. We did not compare with the method in [10] because the tamper detection performance of it was poor for a larger tamper ratio. Several measurements are introduced for quantitative evaluation. (1) Coding Efficiency including code-length (bpp: bit per pixel) and code-quality (PSNR between the reconstructed image and original one); (2) Invisibility including watermark payload (bpp) and the quality of watermarked image(PSNR between the watermarked image and the original one); (3) Tamper detection performance including the probability of false acceptance (PFA) $P_{fa} = \frac{100(N_T - N_{td})}{N_T}\%$, the probability of false rejection (PFR) $P_{fr} = \frac{100N_{ud}}{(N-N_T)}\%$, and the tampering ratio (TR) $r_t = \frac{100N_T}{N}\%$, where N denotes the number of blocks in the test image, N_T denotes the number of tampered blocks, N_{td} denotes the number of tampered blocks which are correctly detected, and N_{ud} denotes the number of valid blocks which are wrongly detected (Note that the block size is 2×2 pixels). (4) Recovery performance (PSNR between the recovered image and the watermarked one).

3.1 Code efficiency and invisibility

Table 2 shows the comparison of coding efficiency for the different images. From Table 2, the code-length of the proposed scheme and [6] are variable for the different image, but that of [13] is fixed. For the proposed method and the method in [6], the smoother the host image is, the smaller the code length is. The code length of the method in [13] is the largest, and that of [6] is the smallest for all images. For the texture images, the code quality of the proposed method is the best. This is due to the fact that the proposed method generates the CC of an 2×2 block with an unfixed length. The block of size 2×2 pixels increases the code length, but improves the performance of tamper detection, evidenced by the following experiments in subsection 3.3.

Since the watermark data are embedded in the LSB planes in the self-embedding watermarking schemes [2], the smaller the watermark payload is, the better the quality of watermarked image is. Table 3 shows the comparison of invisibility for the different images. It is observed from Table 3 that, for the proposed method, the watermark payload for various images ranges from 1.62 bpp to 2.19 bpp, and the PSNRs of the watermarked images range from 39.79 dB to 44.20 dB. The watermark payload and the PSNRs of the method in [6] range from 1.13 bpp to 1.59 bpp and 45.81 dB to 48.59 dB, respectively. In contrast, the watermark payloads of the method in [13] is constant. Hence the PSNRs of the watermarked images by this method are fixed at 37.5 dB, as evidenced in Table 2. Compared Tables 2 with 3, there are the various degree of waste watermark information in [13], and [6], indicated by the difference between the watermark payload and the code length of 0.4 bpp and 0.31 bpp, respectively. In contrast, the watermark payload is identical to the code length in proposed scheme.

Table 2 Comparison of coding efficiency for different images

Images	Code-length (bpp)			Code-quality (dB)		
	Proposed	[13]	[6]	Proposed	[13]	[6]
Boat	1.62	2.6	0.82	35.29	30.51	33.41
Lena	1.63	2.6	0.93	33.04	32.43	33.96
Peppers	1.62	2.6	0.94	32.21	30.72	32.74
Goldhill	1.70	2.6	1.03	31.79	32.69	32.47
Barbara	1.93	2.6	1.06	29.97	28.18	26.57
Man	1.77	2.6	1.09	31.21	28.97	30.44
Flinstones	1.94	2.6	1.14	28.86	23.32	25.19
Baboon	2.19	2.6	1.28	27.48	25.71	25.56

3.2 Security

In the proposed scheme, the CPRS generator defined in [8] is adopted to obtain the pseudorandom sequence $R=\{r_i|i=1,2,\ldots,N\}$, which is used to generate the block-mapping and encrypt the CC of a block. These strategies make the security of the proposed scheme improve greatly.

(1) Watermark confidentiality

If the CC of a block is directly inserted into the LSBs of another block, the four-scanning attack could find out the correlation of blocks [1]. To resist the four-scanning attack, the embedded watermark is the encrypted version of CC in this work. To verify the watermark confidentiality, the different-bit probability in W and W' is defined,

$$p = \frac{\sum_{i=1}^{N}\sum_{m=1}^{v_i}\left|w_{im}-w'_{im}\right|}{\sum_{i=1}^{N} v_i} \tag{20}$$

Where $W=\{w_{im}|i=1,2,\ldots N,\ m=1,..v_i\}$ and $W'=\left\{w'_{im}|i=1,2,\ldots N,\ m=1,..v_i\right\}$ denote the watermark of an image X, which are generated by the different key k and k'

Table 3 Performance comparison of invisibility for different images

Images	Watermark payload (bpp)			Watermarked image quality(dB)		
	Proposed	[13]	[6]	Proposed	[13]	[6]
Boat	1.62	3	1.13	44.20	37.89	48.59
Lena	1.63	3	1.24	44.12	37.92	48.00
Peppers	1.62	3	1.25	43.81	37.92	48.01
Goldhill	1.70	3	1.34	43.20	37.91	47.27
Barbara	1.93	3	1.37	41.73	37.92	47.10
Man	1.77	3	1.40	42.82	37.92	46.90
Flinstones	1.94	3	1.45	41.90	37.81	46.54
Baboon	2.19	3	1.59	39.79	37.92	45.81

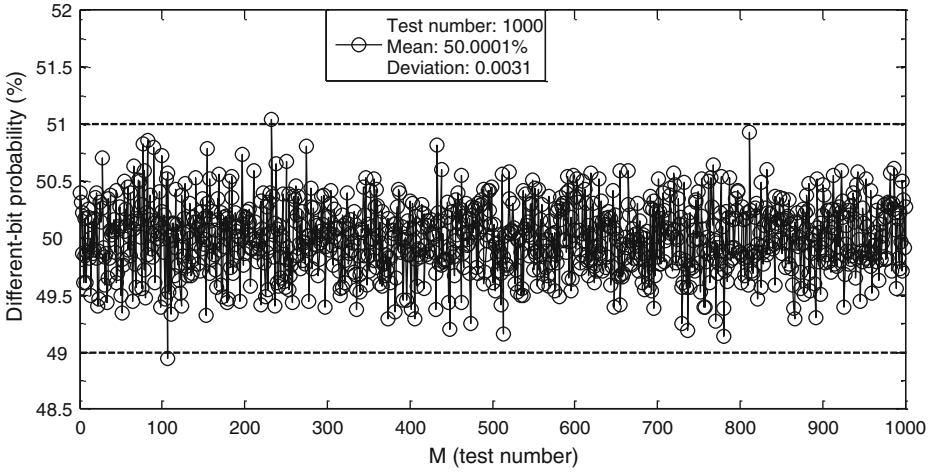

Fig. 3 Distribution of different-bit probability

according to the fore-four steps in sub-section 2.2, respectively. The idea cipher system should be that any tiny changed in secret key lead to the 50 % changing probability for each bit of the encrypted stream [14].

Let M denotes the test number, we can obtain the statistic sequence of different-bit probability $P=\{p_1,p_2,\ldots,p_M\}$ for any two different keys $\{k_j, k_j'\}$ ($j=1,2,\ldots,M$). To valuate distribution characteristics of P, two statistics are defined as follows:

✓ Mean:

$$\overline{P} = \frac{1}{M}\sum_{j=1}^{M} p_j \times 100\% \tag{21}$$

✓ Standard deviation:

$$\Delta P = \sqrt{\frac{1}{M-1}\sum_{j=1}^{M}\left(p_j - \overline{P}\right)^2} \tag{22}$$

Figure 3 shows the distribution of different-bit probability of 1,000 test number. It can be seen from Fig. 3 that the different-bit probability of each test is very close to the theoretical value 50 %. The mean and the standard deviation are 50.0001 % and 0.0031, respectively. These statistical results show that the confusion capability of the proposed method of encrypting the CC is strong and stable.

(2) Block-mapping randomness

As pointed out in [1], the attacker could purposely modify the watermarked images without being detected if he/she obtains the information of the block-mapping sequence in advance. The following experiment examines the distribution characteristic of the watermark embedding position generated by the proposed method.

Let $\delta=N/p$ (N can be divisible by p) be the interval length, the integer interval [1, N] is divided into p small intervals with the same length,

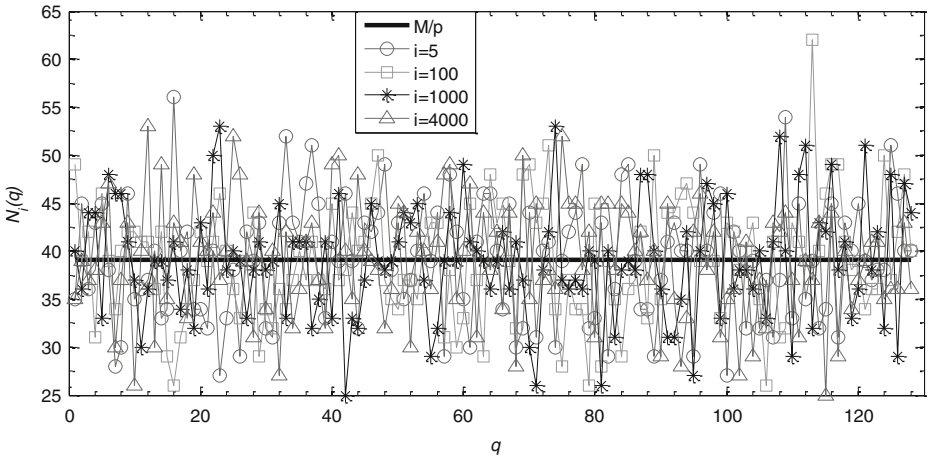

Fig. 4 Distribution of the watermark embedding position

$$[1,N]=\bigcup_{q=1}^{p-1}\left[\delta(q-1)+1,\delta q\right)\bigcup[\delta(p-1)+1,N] \tag{23}$$

Given a secret key K_k (k=1,2,…,M), the index of the mapping block of block X_i, denotes j_k (=$a_i(k)$), is produced by the *step* 5 in sub-section 2.2. We can count the number which the mapping block of block X_i hits the q^{th} small interval [δ(q−1)+1, δq] for different keys, denoted as $N_i(q)$.

$$N_i(q)=\sum_{k=1}^{M}\varpi_i(k) \tag{24}$$

where,

$$\varpi_i(k)=\begin{cases}1, & if\ \lceil a_i(k)/\delta\rceil == q\\ 0, & otherwise\end{cases} \tag{25}$$

Where $\lceil x\rceil$ is the smallest integer more than or equal to x. If the index of the mapping block of each block X_i is random, the theoretical value of $N_i(q)$ should be about M/p for each small interval. Figure 4 shows the distribution of the watermark embedding position of four blocks, where N=4,096, p=128, and M=5,000. It can be seen from Fig. 4 that the distributions for different blocks are similar and the number of hitting each small interval centers on the theoretical value. It indicates that the watermark embedding position of each block is randomly distributed in the whole image based on user key.

3.3 Tamper detection and recovery

To demonstrate the tamper detection performance and recovery quality of the proposed scheme, the Baboon, Flinstones and Lena images with size of 512×512 pixels are chosen. The watermarked Baboon, Flinstones and Lena were generated by the proposed scheme with the same secret key, shown in Fig. 5(a)~(c), with PSNR of 39.79 dB, 41.90 dB and 44.12 dB, respectively. Three tampered images shown in Fig. 5(d)~(f) are described in the following.

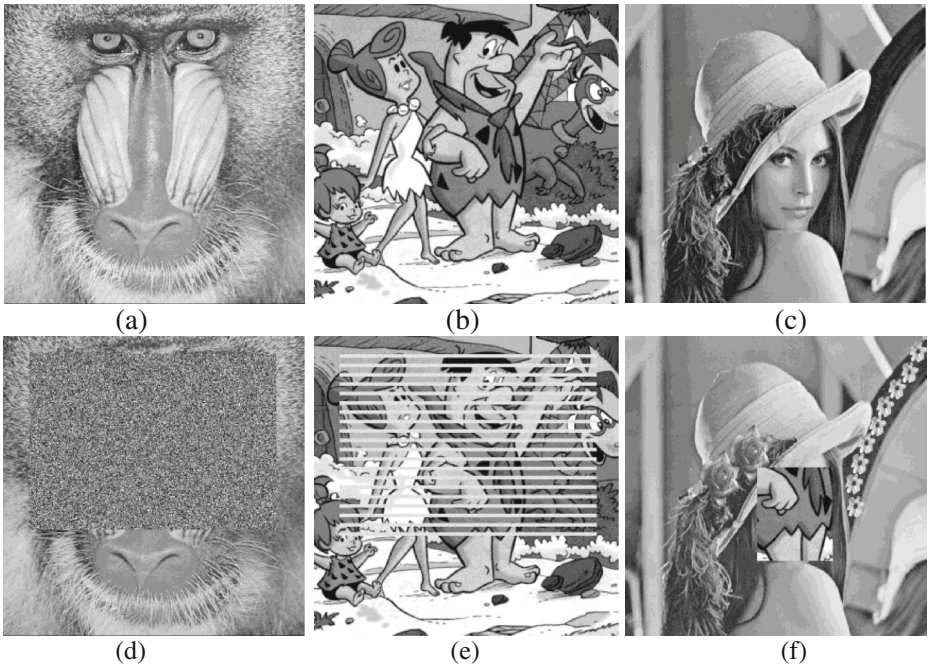

Fig. 5 Watermarked and tampered images **a** watermarked Baboon, **b** watermarked Flinstones, **c** watermarked Lena, **d** tampered Baboon, **e** tampered Flinstones, **f** tampered Lena

✓ Test 1: Figure 5(d) depicts the tampered Baboon image, where the rectangle region of size 300×420 pixels is tampered;

✓ Test 2: Figure 5(e) depicts the tampered Flinstones image. The intensity values of every pixel in the 20 rectangles were replaced with a random integer in the interval [200, 223]. The size of these tampered rectangles is 6×432 pixels, and the distance between any two adjacent rectangles is 10 pixels;

✓ Test 3: Figure 5(f) is the tampered Lena, in which two attacks occurred: Two large flowers and several small ones were pasted, and the face of watermarked Lena replaced with the watermarked Flinstones image of the same region (Collage attack).

Figures 6 and 7 show the tamper detection and recovery results of the three tampered images by the proposed scheme, the methods in [13] and [6], respectively. Table 4 summarizes the quantitative results in terms of TR, PFA, PFR and PSNR.

Test 1 and *Test* 2 are performed to demonstrate the tamper localization accuracy and the quality of recovered images under general tampering. All reported self-recovery schemes can detect the general tampering on a watermarked image. The distinction mainly lies in the tamper localization accuracy. As shown in Table 4, the proposed, Zhang's [13] and Huo's [6] schemes effectively detection any tampered blocks with a probability more than 97 %. However, the methods in [13] and [6] have the large PFR. The PFR of Zhang's [13] and Huo's [6] schemes are up to 37.03 % and 53.95 % as many small regions such as *Test* 2 are tampered. In contrast, the PFR of the proposed scheme is about 3.14 %, evidenced by Fig. 6 (a) ~(f). This is due to the fact that the block size is the 2×2 pixels in the proposed scheme, but the 8×8 pixels in the methods in [13] and [6]. The high PFR transforms to a low quality of recovered images. The larger the PFR is, the worse the quality of the recovered image is,

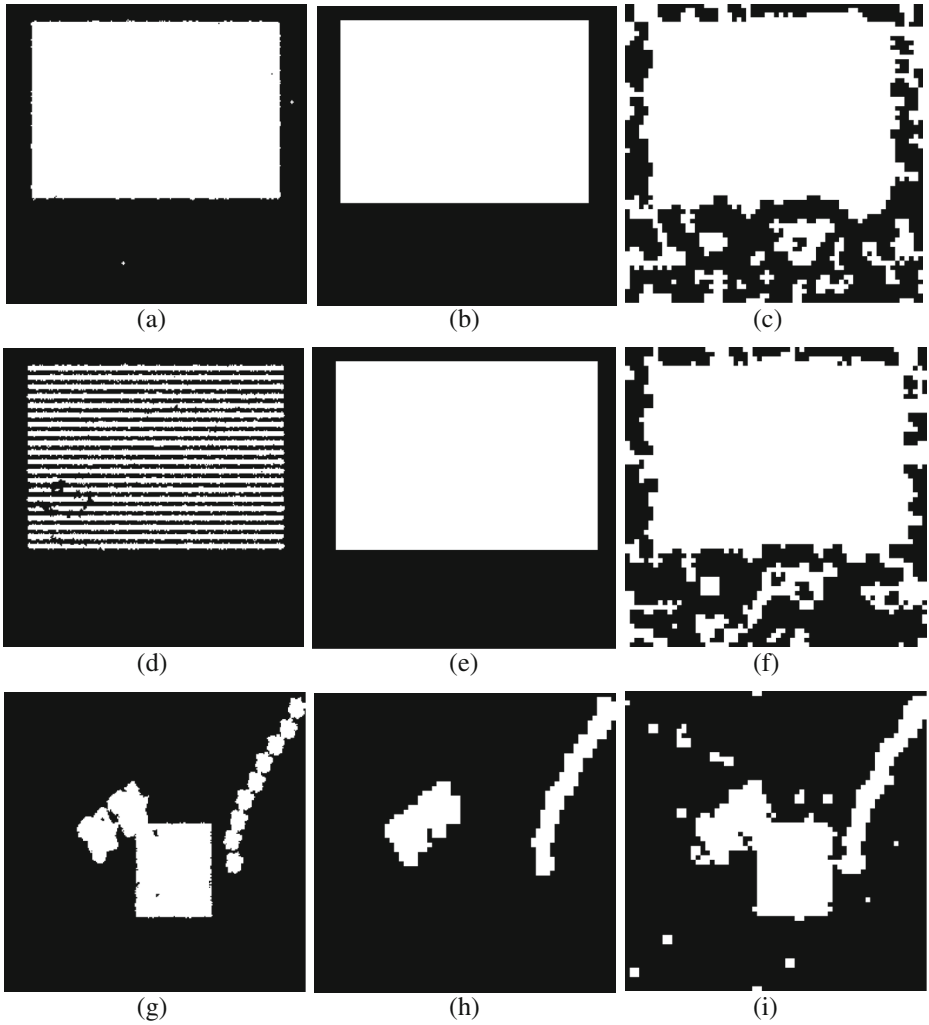

Fig. 6 Tamper detection results. Tampered Baboon **a** proposed **b** [13], **c** [6]; Tampered Flinstones **d** proposed **e** [13], **f** [6]; Tampered Lena **g** proposed **h** [13], **i** [6]

as evidenced by Fig. 7(a) ~(f). The PSNR of the recovered image is 7 dB higher than that of the method in [13], about 12 dB higher than that of the method in [6]. These results indicate that the proposed scheme outperformed the methods in [13] and [6] in tamper detection and recovery under general tampering.

In the *Test* 3, we examine the restoration quality under the malicious counterfeiting attacks including the collage attack. Since the scheme of [13] determines the validity of block by the authentication data embedded in the same block, it is not able to detect the collaged blocks, as shown in Fig. 6(h). It can be seen from Table 4 that the PFA of the method [6] is up to 57.56 % for *Test* 3. The damaged reference-bits could not find the unique solution using the Gaussian elimination method [13] due to the fact that the invalid reference-bits extracted from the collaged blocks were wrongly judged as the valid. As a result, all the tampered regions including the collaged blocks cannot be recovered by the method in [13], as evidenced by

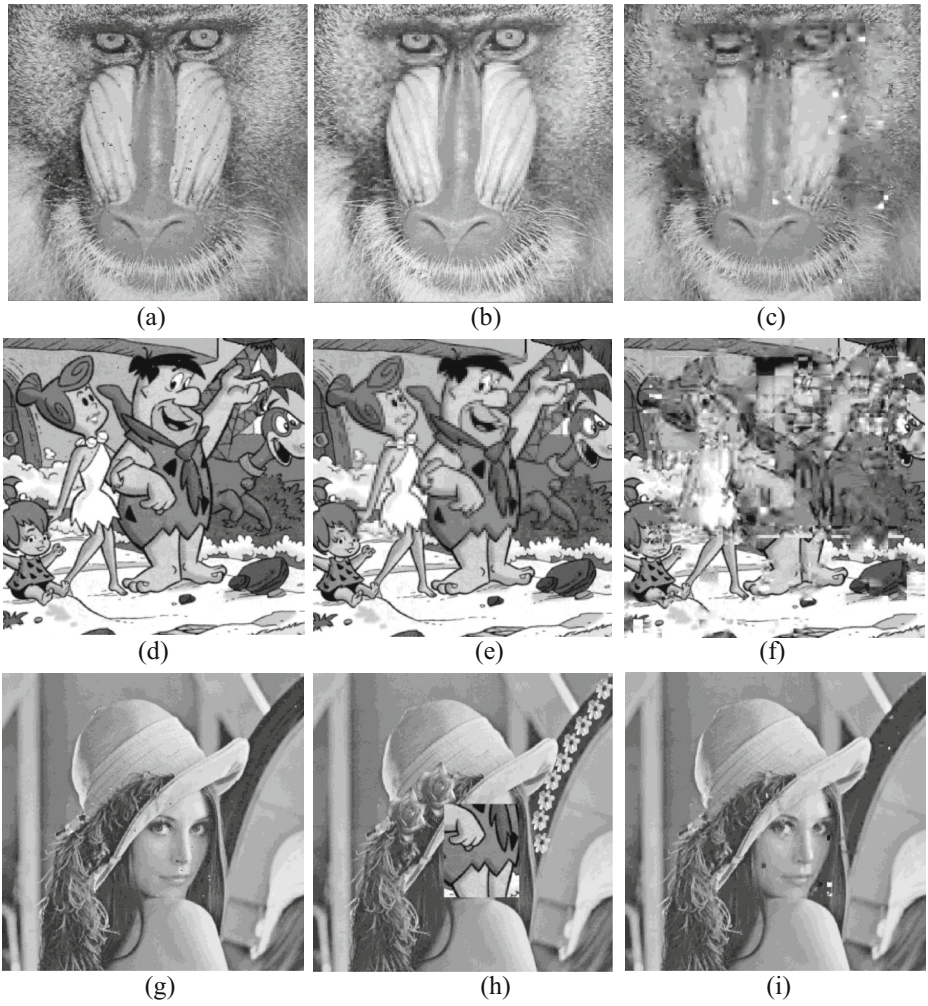

Fig. 7 Recovery results. Tampered Baboon **a** proposed **b** [13], **c** [6]; Tampered Flinstones **d** proposed **e** [13], **f** [6]; Tampered Lena **g** proposed **h** [13], **i** [6]

Fig. 7(h). In contrast, the proposed and Huo's [6] methods could detect and recover all tampered regions. The localization accuracy of the proposed scheme is better than that of the method in [6], as shown in Fig. 6(g) and (i). Accordingly, PSNR of the recovered image by the proposed

Table 4 Performance comparison of tamper detection and recovery

Tests	*TR*	PFA (%)			PFR(%)			PSNR (dB)		
		Our	[13]	[6]	Our	[13]	[6]	Our	[13]	[6]
1	48.07	0.01	0.00	0.93	0.36	4.62	32.94	24.79	23.95	20.74
2	26.48	1.50	0.00	2.34	3.14	37.03	53.95	28.72	21.78	16.13
3	13.58	1.31	57.56	1.48	0.16	2.20	4.89	36.81	18.31	32.42

scheme is 36.81 dB, which is about 4 dB higher than that by Huo's scheme [6], as evidenced by Fig. 7(g) and h(i). These results show that the proposed scheme may achieve more accurate tamper localization accuracy and higher quality of recovered images under the host image tampered by the collage attack.

4 Conclusion

We have presented a self-embedding scheme that generates the embedded data with as few bits as possible while still preserving the superior image recovery quality. The original image is divided into blocks of size 2×2 pixels to improve the accuracy of localization, and a chaotic mapping is adopted to further improve the security. The length of the embedded data varies depending on the complexity of the analyzed block. The watermark payload is minimized and the sufficient information of the image is preserved. Since the embedded data contribute to the content recovery and tamper detection, the recovery quality and security of the watermarking are improved. Future research includes improving the coding efficiency, and extending this approach capable of resisting signal processing operations.

Acknowledgments This work is supported in part by the National Natural Science Foundation of China (60970122, 61170226), the Research Fund for the Doctoral Program of Higher Education (20090184120021), and the Fundamental Research Funds for the Central Universities (SWJTU09CX039, SWJTU10CX09)

References

1. Chang C-C, Fan Y-H, Tai W-L (2008) Four-scanning attack on hierarchical digital watermarking method for image tamper detection and recovery. Pattern Recogn 41:654–661
2. Chen F, He H, Huo Y, Wang H (2011) Self-recovery fragile watermarking scheme with variable watermark payload. Proc Int Conf IWDW LNCS 7128:142–155
3. Fridrich J, Goljan M (1999) Protection of digital images using self-embedding, Proc. Int. Conf. Content Security and Data Hiding in Digital Media
4. Fridrich J, Goljan M, Memon N (2002) Cryptanalysis of the Yeung-Mintzer fragile watermarking technique. J Electron Imaging 11:262–274
5. He H, Zhang J, Tai H-M (2009) Self-recovery fragile watermarking using block-neighborhood tampering characterization. Proc Int Conf Inf Hiding Work LNCS 5806:132–145
6. Huo Y, He H, Chen F (2012) Alterable-capacity fragile watermarking scheme with restoration capability. Opt Commun 285:1759–1766
7. Lee T-Y, Lin SD (2008) Dual watermark for image tamper detection and recovery. Pattern Recogn 41 (2):3497–3506
8. Lian S, Sun J, Wang J, Wang Z (2007) A chaotic stream cipher and the usage in video protection. Chaos Solitons Fractals 34:851–859
9. Qian Z, Feng G, Zhang X, Wang S (2011) Image self-embedding with high-quality restoration capability. Digital Signal Process 21:278–286
10. Qin C, Chang C-C, Chen P-Y (2012) Self-embedding fragile watermarking with restoration capability based on adaptive bit allocation mechanism. Signal Process 92:1137–1150
11. Yang C, Shen J (2010) Recover the tampered image based on VQ indexing. Signal Process 90:331–343

12. Zhang X, Wang S, Qian Z, Feng G (2010) Reversible fragile watermarking for locating tampered blocks in JPEG images. Signal Process 90(12):3026–3036
13. Zhang X, Wang S, Qian Z, Feng G (2011) Reference sharing mechanism for watermark self-embedding. IEEE Trans Image Process 20(2):485–495
14. Zhang J, Wang X, Zhang W (2007) Chaotic keyed hash function based on feedforward-feedback nonlinear digital filter. Phys Lett A 362:439–448

Fan Chen received the M.S. degree in Computer Software and Theory from Chengdu Branch, Chinese Academy Sciences. Currently, he is an associate professor of Southwest Jiaotong University, Chengdu, China. His research interests include multimedia security and digital watermarking.

Hongjie He received her Ph.D. degree in signal and information processing from Southwest Jiaotong University, China. Currently, she is an associate professor of Southwest Jiaotong University, Chengdu, China. Her research interests are in the areas of digital forensics and image processing.

Heng-Ming Tai is a professor in the Department of Electrical Engineering at the University of Tulsa. He received his B.S. degree from National Tsing-Hua University, Taiwan, and his M.S. and Ph.D. degrees from Texas Tech University, all in electrical engineering. His research interests are in the areas of signal and image processing and industrial electronics.

Hongxia Wang born in 1973, Ph. D., professor. Her research interests are in the areas of information hiding, digital watermarking and intelligent information processing.