Multimedia Tools and Applications (2020) 79:1673–1705
https://doi.org/10.1007/s11042-019-08144-5

A tamper detection suitable fragile watermarking scheme based on novel payload embedding strategy

Shiv Prasad[1] · Arup Kumar Pal[1]

Received: 20 March 2018 / Revised: 8 August 2019 / Accepted: 2 September 2019 /
Published online: 7 November 2019

Abstract
Intentional tampering in digital image content is one of the common malpractices in the current digital arena. So in this paper, the authors have proposed a novel fragile watermarking scheme for the localization of tampered image content effectively. Proposed fragile watermarking scheme detects forged image content robustly in block level of two consecutive pixels. In this work, the watermark embedding procedure is comprised of two phases- i.e., authentication code generation from some selected salient bits of each pixel of the original image content, and encryption of the authentication code before realizing it for embedding into the insignificant bits of each pixel in the original cover image. The authentication code is computed from each block using Hamming Code. Subsequently, the encrypted code is concealed into the pixels of that particular block using the suggested payload embedding strategy i.e., pixel adjustment process. The proposed fragile watermarking procedure ensures the high level of security since the encrypted authentication code is embedded into the cover image through an indirect mechanism i.e. block-level pixel adjustment process. This scheme has been implemented and tested on several grayscale images in order to confirm the tampering detection capability against various image manipulation attacks. The experimental results exhibit better performance in terms of various perceptual quality measures like peak signal to noise ratio, structural similarity index, and image fidelity. Further, the presented results demonstrate that the scheme is suitable for detecting whether a concerned image has undergone any form of tampering or not and achieves standard results in terms of false-positive rate, false-negative rate, true positive rate, tamper detection accuracy, and normalized cross-correlation.

Keywords Fragile watermarking · Hamming codes · Logistic-map · Tamper detection

✉ Shiv Prasad
psad.shiv@gmail.com

Arup Kumar Pal
arupkrpal@gmail.com

[1] Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand 826004, India

1 Introduction

In the present era, the people are globally connected through various platforms like social media, e-healthcare, e-commerce, e-business, etc. where they share information in the form of different multimedia components. In social media, people share their private data quite easily at any moment and from any location. In e-healthcare based applications, patients are uploading their medical report to receive an expert opinion from any specialized hospitals. Similarly, nowadays, the availability of any particular product is not a big concern due to the advancement of e-commerce and e-business based applications. In all these applications, the gigantic volume of digital data is exchanged through the Internet. However, the Internet itself is not a protected or secured communication medium, so fraudulent users may effortlessly intercept the transmitted data. As a result, security is an integral part of the transmission of secret data. Different security mechanisms have been adopted to protect the security of essential data in terms of confidentiality [23], integrity [23], and authentication [23]. These security services are achieved in different ways like (a) cryptography and steganography for protecting the confidentiality property, (b) cryptographic hash function and fragile watermarking for ensuring the integrity, and (c) digital signature and robust watermarking for holding the authentication property. In this paper, we have addressed the integrity issue in digital images, which are considered as the most widely used multimedia component. In fragile watermarking, the authentication code is computed from some selective components of an image, and thereafter the authentication code is concealed into non selected components of the image to form a watermarked image [10]. In the integrity verification process, the authentication code is computed similarly from the watermarked image. Subsequently, the embedded code is extracted from the watermarked image. Both the codes will be the same if the watermarked image is not manipulated or altered by malicious users. In this paper, we have suggested a novel fragile watermarking scheme with the intention to improve the visual quality of the watermarked image as well as to realize the high level of tamper detection capability. In this work, the payload embedding is carried out based on the suggested embedding approach, i.e., block-level pixel adjustment process (BPAP). The proposed work is presented in two aspects, i.e., Hamming code-based authentication code generation and later concealing the encrypted authentication code into the cover image by BPAP approach.

In literature, a number of the digital image based fragile watermarking schemes are found to protect the integrity property of the image content. He et al. [6] have suggested a block level fragile watermarking method, which works on non-overlapping blocks of size 8×8 pixels. The watermark or authentication codes are primarily computed from each block, and later, the same is hidden randomly into the LSB position of the pixels in original image based on the secret key to protect the embedded payload from various attacks. However, the main limitation of their work is that the tampered region is identified on the block of size 8×8 (i.e., 64 pixels) even if only a single tampered pixel is found. However, Chang et al. [4] have proposed a fragile watermark embedding process, which divides the cover image into non-overlapping blocks of size 2×2. In [7, 8] have suggested an image tamper detection scheme, which divides the digital image into two categories as edge blocks, and non-edge blocks. Watermark is embedded into the digital image based on the characteristics of edge blocks and non-edge blocks and their approaches are found effective to improve the authentication of the digital image.

Several fragile watermarking schemes have incorporated a number of chaotic maps [3, 5, 19, 21] to enhance the security level. Chang et al. [3] have suggested a two-pass logistic-map combining with hamming code to detect tampering in the digital image. The two pass logistic-map contains a

R_i	R_1	R_2	R_3	R_4	R_5	R_6
Range	[0 - 7]	[8 - 15]	[16 - 31]	[32 - 63]	[64 -127]	[128-255]
Width	8	8	16	32	64	128
Payload/Capacity (bits)	3	3	4	5	6	7

Fig. 1 The quantization range table

private key to resist vector quantization attack. Tong et al. [21] have suggested a fragile watermarking scheme for localization of tampered areas effectively in the digital image which presents the combination of the most significant bit and the least significant bit (LSB) information to embed the payload into the cover image. Chen et al. [5] have presented a self-embedding fragile watermarking scheme using pseudorandom chaos sequence, which is found proficient against various image tampering attacks. Recently, Sreenivas and Kamakshiprasad [19] have proposed an image tamper localization technique, which generates watermark bits or authentication code from 2×2 block of the digital image using the chaotic maps and randomly embeds the authentication code or watermark bits into selected distinct blocks. In [20], the authors have computed watermark bits from the first 5 MSBs of each pixel, and later first 3 LSBs of the corresponding pixel are replaced by the watermark bits information. Their scheme also identifies the tampered region successfully. In [11], Nazari et al. have devised another chaotic map based fragile watermarking scheme where the authentication code bits are not fixed throughout the blocks. This adaptive payload embedding approach improves the visual quality of the watermarked image. Trivedy and Pal [22] have presented a fragile watermarking scheme based on pixel-level using logistic-map. The logistic-map generates a chaotic sequence that is used in the key matrix formation, and subsequently the key matrix is realized in the watermark embedding process. Since their scheme is based on the secret key matrix, so it is secure, but it has high computational overhead. Rawat and Raman [17] have developed another chaos based fragile watermarking scheme by employing two chaotic maps, and the initial values are sensitive to the chaotic maps. The parameters of chaotic maps are used as the secret key. This process improves the security-level of the watermark. Researchers also devise several fragile watermarking schemes based on Hamming code [2, 3]. Chana et al. [2] have presented an image authentication using the combination of the Hamming code technique, Torus automorphism, and bit rotation technique to identify tampering effectively. This hybrid scheme can efficiently remove burst bit errors, and recover the pixels. In [13, 14, 18], the common attacks on watermarked images are simply influenced by the eavesdropper. In this way, the tampering is

R_i	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
Range	[0 - 7]	[8 - 15]	[16 - 23]	[24 - 31]	[32 - 39]	[40 - 47]	[48 - 55]	[56 - 63]
Width	8	8	8	8	8	8	8	8
Payload/Capacity (bits)	3	3	3	3	3	3	3	3

Fig. 2 The new quantization range table

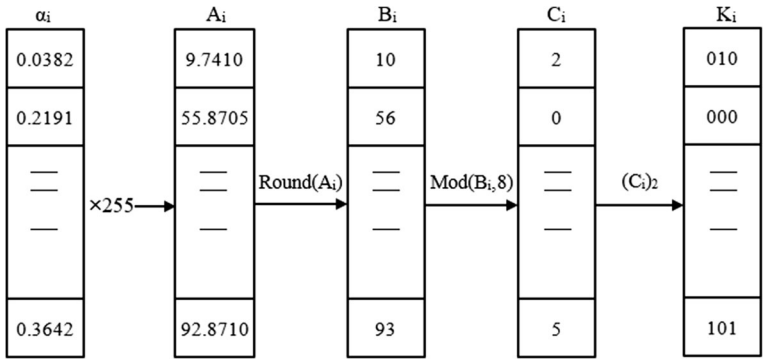


Fig. 3 Process of binary secret key generation

not visually recognizable. This type of attacks causes serious issues in the medical data [1, 25] where the proper diagnosis depends especially on correct medicals reports. Recently, Bravo-Solorio et al. [1] have proposed a fragile watermarking scheme where the computation overhead of the payload embedding process is reduced. Peng et al. [12] have presented an image authentication technique, which is implemented with two identical host images. Their scheme embeds the secret data into one host image and the some supplementary data is embedded into another host image. Yin et al. [26] have proposed the reversible image authentication scheme based on the Hilbert Curve mapping, where pixels are first mapped to a one-dimensional vector using the Hilbert Curve, and divided into non-overlapping sets. The reversible secret data embedding technique embeds the authentication codes into each non-overlapping set. Qin et al. [15] have suggested a novel self-embedding fragile watermarking proposal based on vector quantization (VQ) and index contribution where the embedding of watermark bits depend on hash-bits generation. Later, Qin et al. [16] have given another improved fragile watermarking method based on overlapping embedding strategy with high-quality recovery capability. It is also found effective during localization of tampered regions. Lo et al. [9] have proposed a reversible image authentication scheme for digital images. The watermark or authentication codes are embedded into the cover image, which is generated by the selected random number seed.

The above discussed fragile watermarking schemes are basically suitable for localizing the tampered regions if any image alteration happens intentionally/unintentionally. In this paper, we have suggested another fragile watermarking scheme with enhanced properties. In this contributed work, the authentication code is generated from the most significant bits of pixels using Hamming code, and later the authentication code is further encrypted by secret bit streams which are generated using logistic-map. Further, the encrypted code is embedded into the least significant bits of pixels using the novel BPAP. The indirect payload embedding process is more suitable for retaining the high visual quality of the watermarked image. The above procedure presents an excellent secure fragile watermarking scheme, as well as exhibits the decent visual quality of watermarked images. The main contributions of the proposed work are as follows:

- Proposed scheme enhances the level of security of a fragile watermarking scheme since the authentication code is generated using the Hamming code and logistic-map.
- This approach increases the level of visual quality compared to other fragile image watermarking schemes because the watermark embedding process is carried out using the suggested BPAP.

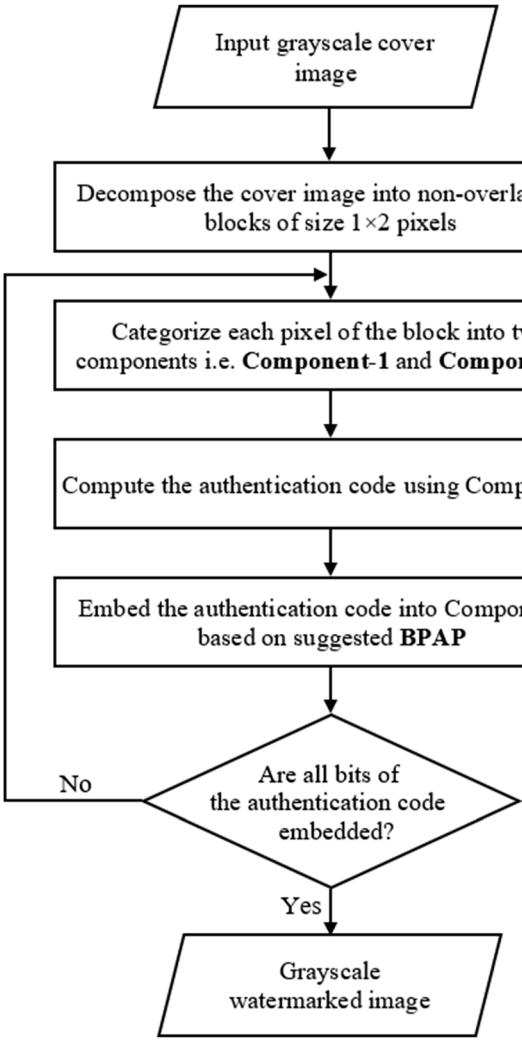


Fig. 4 The major steps of the watermark embedding procedure

- Proposed scheme effectively locates the tampered regions if any alteration occurred in the image content.
- Proposed procedure is effective and acceptable in terms of various perceptual quality measures.

The rest of the paper is presented as follows. In section 2, the related preliminaries are briefly discussed. The proposed fragile digital image watermarking scheme is presented in Section 3, where watermark embedding, watermark extraction, and tamper detection procedures are elaborated in three different sub-sections. The experimental results related to watermark embedding, watermark extraction, and tamper detection procedure are demonstrated in Section 4. Finally, the conclusion of the paper is drawn in Section 5.

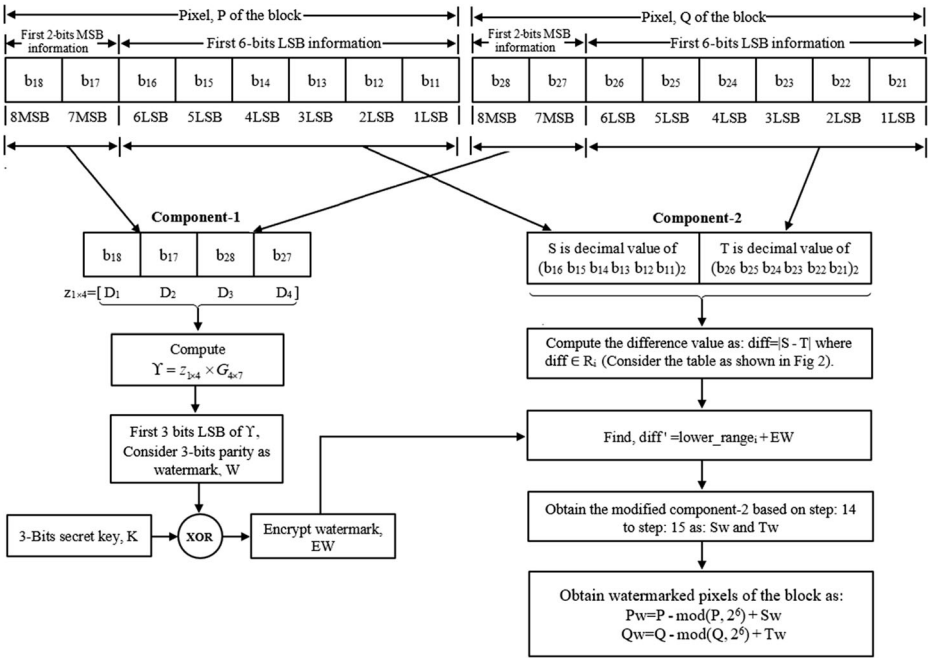


Fig. 5 The details steps of the watermark embedding procedure

2 Preliminaries

In this section, the logistic map based key sequence generation, and (7, 4) Hamming codes are discussed briefly. These are used in the proposed work for generating secure watermark bits. Subsequently, the suggested BPAP is based on pixel-value differencing (PVD) concept, so the working process of PVD is also described briefly.

2.1 Logistic map based key sequence

The logistic map based key sequence is commonly used in various security mechanisms. It has wide applications in the field of watermarking. A secret key sequence can be produced by considering the seed values of the logistic map. In this work, a secret key matrix is derived using the logistic map. It effectively generates a random sequence using the Eq. 1 and Eq. 2 respectively. The initial parameters like β and α_0 are considered as seed values. The subsequently generated sequences are as follows:

$$\alpha_1 = \beta \times \alpha_0 \times (1-\alpha_0) \tag{1}$$

$$\alpha_{(i)} = \beta \times \alpha_{(i-1)} \times \left(1-\alpha_{(i-1)}\right) \tag{2}$$

The sender and the receiver are able to produce the same sequence if they are using the same seed values i.e., β and α_0.

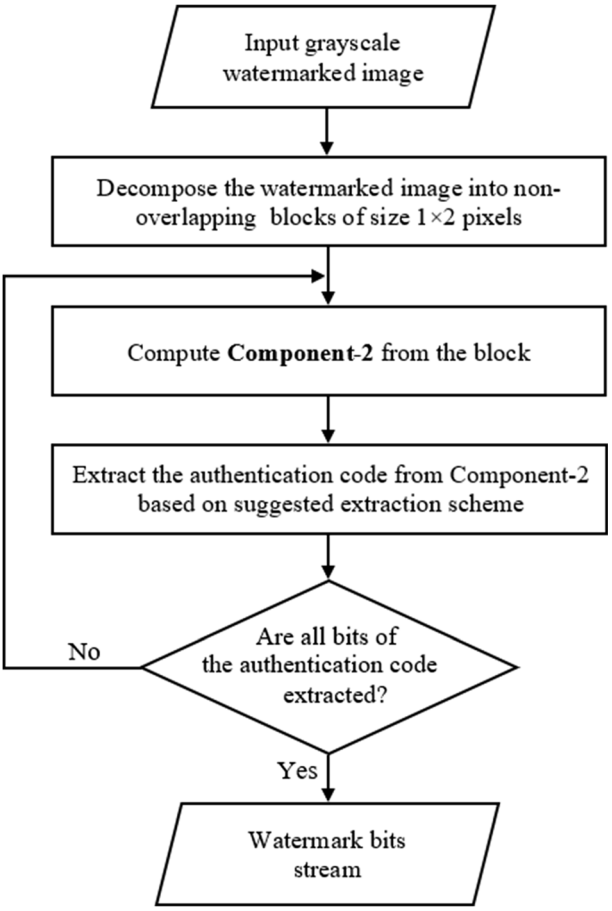


Fig. 6 The overview of watermark extraction procedure

2.2 Hamming code

In coding theory, the Hamming code is one of the popular linear error correcting codes. This section presents the (7, 4) Hamming code which has four data bits and adds three parity bits. Hamming codes are generated by multiplying the information bits with the generator matrix.

The systematic generator matrix for the (7, 4) Hamming code is given below:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{3}$$

Also, the parity-check matrix in the systematic form is as follows:

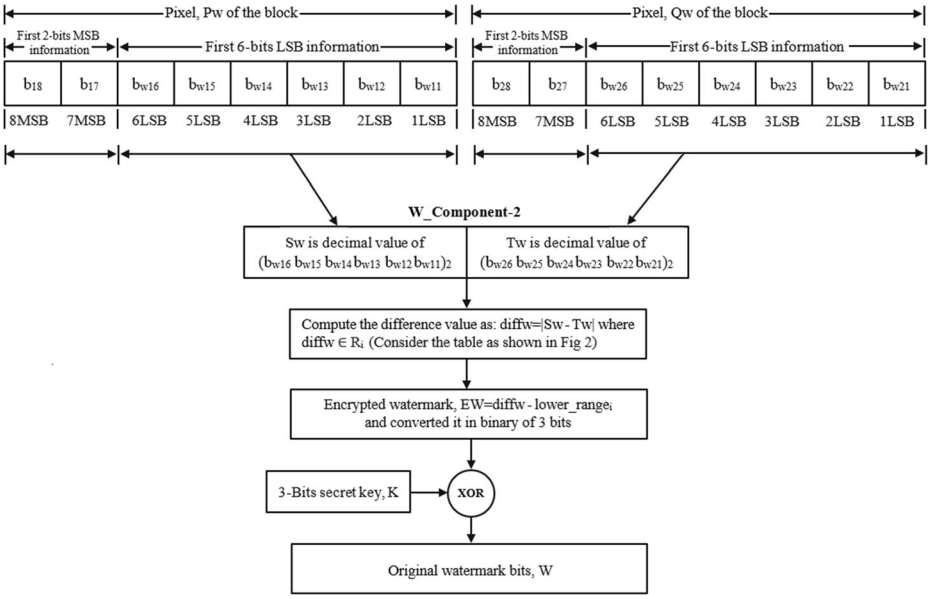


Fig. 7 The detail watermark extraction procedure

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

2.3 Pixel-value differencing (PVD)

The pixel-value differencing (PVD) [24] is one of the indirect data hiding approaches. In this mechanism, initially the cover image is decomposed into the non-overlapping block of two consecutive pixels i.e., a block of size 1×2 pixels. Subsequently, the secret message bits are concealed into each block based on the nature of pixels Q_i and Q_{i+1} of each block. The difference value, $diff_i$ between two pixels is computed by $diff_i = |Q_i - Q_{i+1}|$. The $diff_i$ value is further quantized into several regions, as shown in Fig. 1. Each region is identified by the lower and upper bound values i.e., $[lower_i, \ upper_i]$. The number of payload bits (k) in each block depends on the quantization range table and it is determined as $k = \lfloor \log_2(upper_i - lower_i + 1) \rfloor$. The obtained bit sequence is converted into decimal value (k_d). Later, a new difference value $\left(diff'_i\right)$ is obtained as $diff'_i = k_d + lower_i$.

The modified pixel values are computed based on the following Eq.5.

$$\left(Q'_i, Q'_{i+1}\right) = \begin{Bmatrix} (Q_i + \lceil l/2 \rceil, Q_{i+1} - \lfloor l/2 \rfloor), if\ Q_i \geq Q_{i+1}\ and\ diff'_i > diff_i \\ (Q_i - \lfloor l/2 \rfloor, Q_{i+1} + \lceil l/2 \rceil), if\ Q_i < Q_{i+1}\ and\ diff'_i > diff_i \\ (Q_i - \lceil l/2 \rceil, Q_{i+1} + \lfloor l/2 \rfloor), if\ Q_i \geq Q_{i+1}\ and\ diff'_i \leq diff_i \\ (Q_i + \lceil l/2 \rceil, Q_{i+1} - \lfloor l/2 \rfloor), if\ Q_i < Q_{i+1}\ and\ diff'_i \leq diiff_i \end{Bmatrix} \tag{5}$$

where, $l = \left|diff'_i - diff_i\right|$.

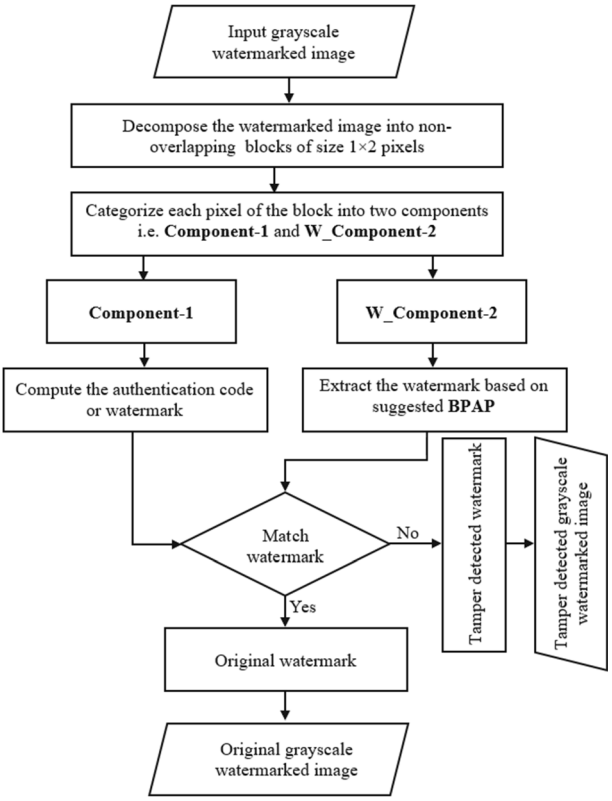


Fig. 8 The overview tamper detection procedure

Afterwards, pixels Q_i and Q_{i+1} of each block are replaced by the stego-pixels $Q_i^{'}$ and $Q_{i+1}^{'}$. The receiver will find the difference of *i-th* block $diff_i^{'} = \left|Q_i^{'} - Q_{i+1}^{'}\right|$. The difference $diff_i^{'}$ is used to search the number of concealing bit streams in *i-th* block using the quantization range given in Fig. 1. Finally, the secret bit streams are obtained after converting the decimal value of $\left(diff_i^{'} - lower_i\right)$ into binary form. An example of the PVD process is illustrated below.

Example 1 The embedding procedure is illustrated based on consideration of two consecutive pixels i.e., 102 and 120; and the secret message bits stream i.e., $(1011)_2$.

Solution Calculate $diff_i = |120-102| = 18$, it decides the secret message bits embedding capacity/payload into two consecutive pixels.

$$k = \lfloor \log_2(upper_i - lower_i + 1) \rfloor = \lfloor \log_2(31-16+1) \rfloor = \left\lfloor \log_2\left(2^4\right) \right\rfloor = 4bits$$
$$diff_i^{'} = lower_i + \text{Secret message } (In \quad decimal \quad representation)$$
$$diff_i^{'} = 16 + 11 = 27 \ \ and \ \ l = \left|diff_i^{'} - diff_i\right| = |27-18| = 9$$

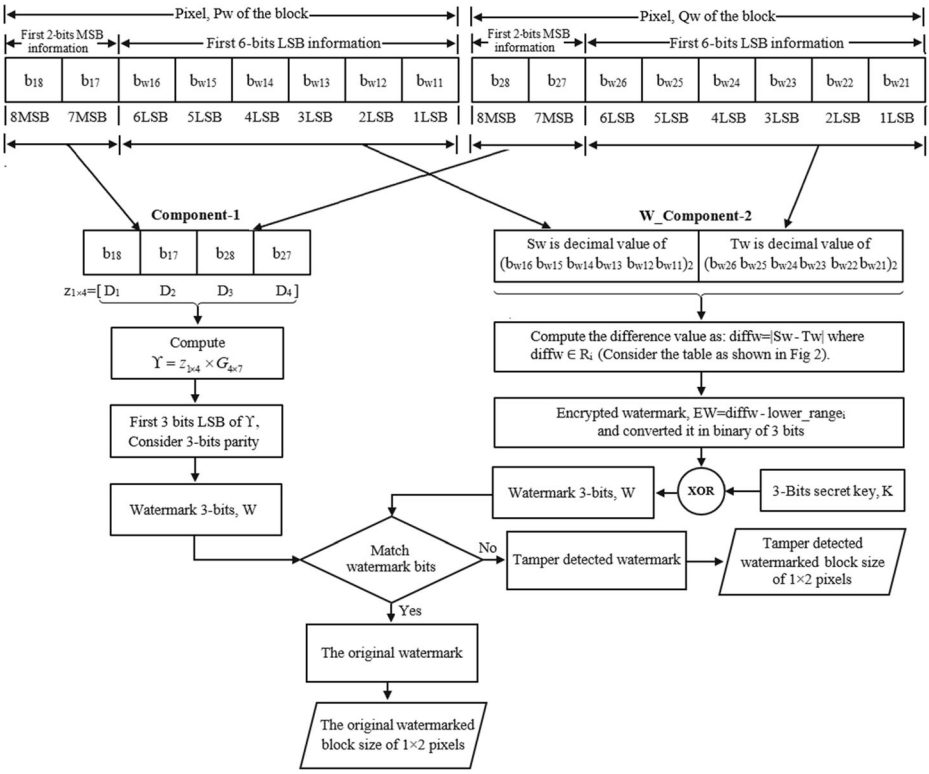
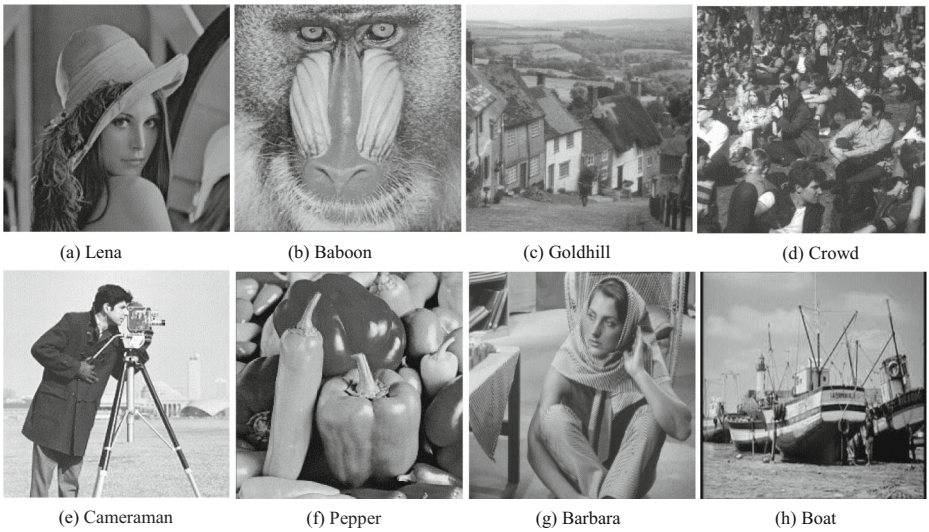


Fig. 9 The details of tamper detection procedure

Fig. 10 (a-h) Original cover images of size 512 × 512



(a) Lena (b) Baboon (c) Goldhill (d) Crowd

(e) Cameraman (f) Pepper (g) Barbara (h) Boat

Fig. 11 (a-h) Watermarked images based on the proposed methodology

The modified new pixel values are calculated using Eq. 5.

$$\left(Q_i^{'}, Q_{i+1}^{'}\right) = (102-4, 120+5)\ if\ 102 < 120\ and\ 27 > 18$$
$$\left(Q_i^{'}, Q_{i+1}^{'}\right) = (98, 125).$$

3 Proposed scheme

In this scheme, the authors have proposed a secure fragile watermarking scheme, where the watermark bits are embedded into a grayscale cover image based on the suggested *BPAP*. In this work, the authors have considered a new quantization range table for *BPAP*. The newly modified quantization range table has eight regions: i.e., R_1, R_2, R_3, R_4, R_5, R_6, R_7, and R_8 respectively. The embedding capacity of each region has a constant size of three bits. The newly quantization range table is shown in Fig. 2.

Table 1 Experimental results of the proposed scheme

Image Name (512 × 512)	Proposed Scheme			
	Payload (bits)	PSNR (dB)	SSIM	IF
Lena	393,216	42.01	0.9993	0.9837
Baboon	393,216	42.29	0.9994	0.9849
Goldhill	393,216	42.31	0.9995	0.9850
Crowd	393,216	41.85	0.9994	0.9829
Cameraman	393,216	41.62	0.9991	0.9824
Pepper	393,216	42.21	0.9991	0.9837
Barbara	393,216	42.29	0.9995	0.9849
Boat	393,216	42.11	0.9997	0.9843

Table 2 Comparison in terms of *PSNR* values

Image Name (512 × 512)	Chang et al. [3] PSNR (dB)	Tong et al. [21] PSNR (dB)	Nazari et al. [11] PSNR (dB)	Sreenivas et al. [19] Method-A PSNR (dB)	Proposed Scheme PSNR (dB)
Lena	37.76	37.57	35.79	37.98	42.01
Baboon	37.87	38.14	35.65	37.92	42.29
Goldhill	37.91	37.89	36.95	37.95	42.31
Crowd	37.96	37.38	36.91	38.03	41.85
Cameraman	37.87	37.19	37.45	37.89	41.62
Pepper	37.85	37.65	36.23	37.98	42.21
Barbara	37.90	37.91	36.31	37.93	42.29
Boat	37.91	37.82	36.69	37.93	42.11

Later, a logistic map generates the chaotic sequence values using Eq. 1 and Eq. 2. The sequence values are found in the range from 0 to 1. This sequence is converted into a binary number of 3 bits representation based on the Eq. (6–9). An illustration of the process of generating the binary secret key sequence is depicted in Fig. 3 where it shows the mapping process of a logistic map based generated real value into corresponding 3 bits binary data.

$$A_i = \alpha_i \times 255 \tag{6}$$

$$B_i = round(A_i) \tag{7}$$

$$C_i = \mathrm{mod}(B_i, 8) \tag{8}$$

$$K_i = (C_i)_2 \tag{9}$$

The above mentioned newly quantization range table and the logistic map based generated secret key sequence are considered in the watermark embedding procedure, the watermark extraction procedure, and the tamper detection procedure, respectively. In the following subsections, we have demonstrated the working procedure of each process with suitable diagrams along with algorithmic steps.

Table 3 Comparison in terms of *SSIM* values

Image Name (512 × 512)	Chang et al. [3] SSIM	Tong et al. [21] SSIM	Nazari et al. [11] SSIM	Sreenivas et al. [19] Method-A SSIM	Proposed Scheme SSIM
Lena	0.9793	0.9842	0.9989	0.9410	0.9993
Baboon	0.9936	0.9951	0.9920	0.9737	0.9994
Goldhill	0.9889	0.9916	0.9928	0.9589	0.9995
Crowd	0.9876	0.9906	0.9912	0.9877	0.9994
Cameraman	0.9713	0.9759	0.9789	0.9716	0.9991
Pepper	0.9834	0.9803	0.9971	0.9950	0.9991
Barbara	0.9874	0.9904	0.9979	0.9523	0.9995
Boat	0.9834	0.9884	0.9935	0.9467	0.9997

Table 4 Comparison in terms *IF* values

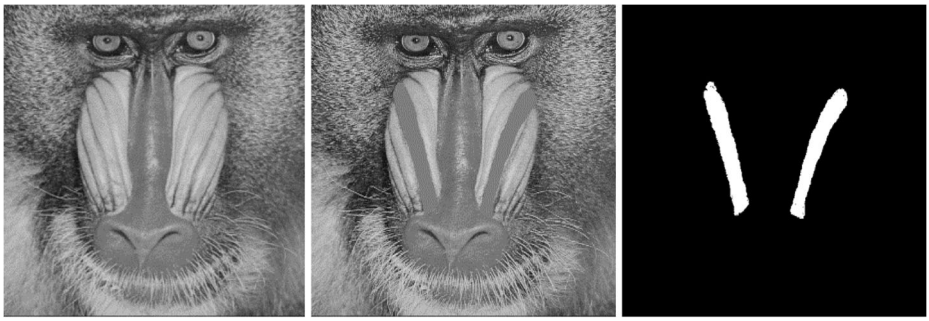
Image Name (512 × 512)	Chang et al. [3] IF	Tong et al. [21] IF	Nazari et al. [11] IF	Sreenivas et al. [19] Method-A IF	Proposed Scheme IF
Lena	0.9579	0.9547	0.9451	0.9595	0.9837
Baboon	0.9588	0.9609	0.9498	0.9589	0.9849
Goldhill	0.9588	0.9586	0.9485	0.9592	0.9850
Crowd	0.9593	0.9535	0.9481	0.9600	0.9829
Cameraman	0.9584	0.9513	0.9542	0.9586	0.9824
Pepper	0.9556	0.9536	0.9447	0.9588	0.9837
Barbara	0.9587	0.9587	0.9486	0.9590	0.9849
Boat	0.9555	0.9545	0.9521	0.9590	0.9843

3.1 Watermark embedding procedure

In this procedure, we have considered the grayscale cover image of *256* intensity levels, and in the watermark embedding process, each pixel is converted into eight bits of binary data representation. The eight bits binary data are divided into two categories, where in the first category, two bits of binary data is obtained from 1st position *MSB* content and 2*nd* position *MSB* content. In the second category, first six *LSB* bits (i.e. information from 6^{th}*LSB,* 5^{th}*LSB,* 4^{th}*LSB,* 3^{rd}*LSB,* 2^{nd}*LSB,* and *1stLSB*) are stored collectively. In this paper, the first category is represented as *Component-1*, and the second one is denoted as a *Component-2*. Here, we have formed 4 bits of information using *Component-1* from a block where each block contains two pixels. Three parity bits is obtained from 4 bits of information using the Hamming code. The 3-bit parity is considered as watermark or authentication code and subsequently, these bits are encrypted after performing *XOR* operation using logistic map based generated secret 3 bits binary sequence. The encrypted watermark bits are embedded into *Component-2* of that particular block. The embedding process is done using the suggested *BPAP* where our main concern is to retain the high visual quality of the watermarked image. The major steps of the watermark embedding procedure is shown diagrammatically in Fig. 4, and the details steps of the watermark embedding process is depicted with an example in Fig. 5. Further, the watermark embedding procedure of the proposed work

(a) (b) (c)

Fig. 12 **a** The watermarked Lena, **b** Tampered Lena, and **c** Actual tampered areas



(a) (b) (c)

Fig. 13 **a** The watermarked Baboon **b** Tampered Baboon, and **c** Actual tampered areas

is presented in Algorithm 1.

Example 2 An illustration for embedding of the watermark is presented here. Suppose 155 and 120 are two consecutive non-overlapping pixels of a block from the grayscale cover image and a secret key, K is $(101)_2$.

Solution The first cover pixel, $P = (155)_{10} = (10\underline{011011})_2$ and adjusted pixel value is S=*Component-2* in decimal (i.e. information bits from first six LSB position of first pixel, P), so $S = (\underline{011011})_2 = (27)_{10}$ and x = *Component-1* (i.e. information bits from 1st and 2nd position MSB content of P) $= (10)_2$. Similarly, for the second cover pixel, $Q = (120)_{10} = (01\underline{110\,0\,0})_2$ find T=*Component-2*=$(\underline{110\,0\,0})_2 = (56)_{10}$ and y = *Component-1* = $(01)_2$ are selected from two consecutive non-overlapping pixels P and Q of a block.

Now, four data bits are $z = x \| y = (1\ 0\ 0\ 1)_2$

$$\Upsilon = z_{1\times 4} \times G_{4\times 7} = [1 \quad 0 \quad 0 \quad 1] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = \left(1001\ \underline{1\,0\,0}\right)_2.$$

(a) (b) (c)

Fig.14 **a** The watermarked Goldhill **b** Tampered Goldhill, and **c** Actual tampered areas

Input: Grayscale cover image, Seed values of Logistic map
Output: Grayscale watermarked image.

Begin

1. Read a grayscale cover image.
2. Decompose the cover image into non-overlapping blocks of size *1×2*.
3. Read two consecutive pixels i.e., *P* and *Q* from each block and convert each pixel into eight bits of binary representation.
4. Process each block by selecting *S=Component-2* in decimal (i.e., information bits obtained from first six LSB position of first pixel, *P*) and *x=Component-1* in binary representation (i.e., information bits obtained from 1st and 2nd position MSB content of *P*). Similarly, for the second pixel (*Q*), find *T=Component-2* in decimal form and *y= Component-1* in binary representation.
5. Append *x* and *y*, i.e. *z=x||y* to obtain the data in 4-bits binary representation as: $z_{1\times4}=[D_1\ D_2\ D_3\ D_4]$.
6. Compute the Hamming code using the symmetric Hamming generator matrix (*G*): $\Upsilon = z_{1\times4} \times G_{4\times7}$ and the 3-bits parity is obtained from the first 3 bits *LSB* of the Hamming code
7. Consider the parity bits as watermark, *W*.
8. Based on the seed values of logistic map, find the secret key K using Eq.6 to Eq.9
9. Encrypt the 3-bit watermark with binary 3-bit secret key using *XOR* operation as follow: $EW = W \oplus K$
10. Compute the absolute difference value between *S* and *T* as $diff = |S - T|$
11. Estimate the $lower\_range_i$ if the $diff$ belong to R_i (Consider the table as shown in Fig 2).
12. Find $diff' = lower\_range_i + EW$ (In decimal value)
13. Compute the difference value: $l = |diff - diff'|$
14. Obtain the modified *Component-2* as *Sw* and *Tw* based on the following equation:

$$(Sw, Tw) = \begin{Bmatrix} \left(S + \lceil l/2 \rceil, T - \lfloor l/2 \rfloor\right), \text{ if } S \geq T \text{ and } diff' > diff \\ \left(S - \lfloor l/2 \rfloor, T + \lceil l/2 \rceil\right), \text{ if } S < T \text{ and } diff' > diff \\ \left(S - \lceil l/2 \rceil, T + \lfloor l/2 \rfloor\right), \text{ if } S \geq T \text{ and } diff' \leq diff \\ \left(S + \lceil l/2 \rceil, T - \lfloor l/2 \rfloor\right), \text{ if } S < T \text{ and } diff' \leq diff \end{Bmatrix}$$

15. Perform the readjustment process in underflow, and overflow conditions as follows:

If $(Sw < 0 \,||\, Tw < 0)$

$$(Sw,\ Tw) = \left\{\left(Sw + \lceil l/2 \rceil\right), \left(Tw + \lceil l/2 \rceil\right)\right\}$$

Else if $(Sw > 63 \ ||\ Tw > 63)$

$$(Sw,\ Tw) = \left\{\left(Sw - \lceil l/2 \rceil\right), \left(Tw - \lceil l/2 \rceil\right)\right\}$$

If End

16. Obtain the watermarked pixels as

$$Pw = P - mod\left(P, 2^6\right) + Sw$$

$$Qw = Q - mod\left(Q, 2^6\right) + Tw$$

17. Process rest of the blocks using steps(3-16).
18. Construct the digital watermarked image, WI.

End:

(a) (b) (c)

Fig. 15 **a** The watermarked Crowd, **b** Tampered Crowd, and **c** Actual tampered areas

3-bit parity $(100)_2$ is obtained from the first 3 bits LSB of Υ. Consider the parity bits $(100)_2$ as the authentication code or watermark bits (W). The 3-bit watermark, W is encrypted with the 3-bits secret key, $K = (101)_2$ using XOR operation as: $EW = W \oplus K = (100)_2 \oplus (101)_2 = (001)_2 = (1)_{10}$. Now, compute the difference value of S and T based on $diff = |S - T| = |27 - 56| = 29$. The $diff$ value belong to R_4 in Fig. 2. Later, compute $diff' = lower\_range_i + EW = 24 + 1 = 25$. Find the difference value, $l = |diff - diff'| = |29 - 25| = 4$. Considering the case : $S < T$ *and* $diff' < diff$.

We have obtained as $(Sw, Tw) = (S + \lceil l/2 \rceil, T - \lfloor l/2 \rfloor) = (27 + 2,\ 56 - 2) = (29,\ 54)$.

The original watermarked pixels are obtained based on the following process

$$Pw = P - mod\left(P, 2^6\right) + Sw = 155 - mod\left(155, 2^6\right) + 29 = 157.$$
$$Qw = Q - mod\left(Q, 2^6\right) + Tw = 120 - mod\left(120, 2^6\right) + 54 = 118.$$
$$(Pw, Qw) = (157, 118).$$

Finally, watermarked pixels are 157, and 118.

3.2 Watermark extraction procedure

In this subsection, the watermark extraction procedure is discussed. Figure 6 shows the brief outline of the watermark extraction process.

In this process, the *Component-2* of each block is considered to attain the watermark bits.

(a) (b) (c)

Fig. 16 **a** The watermarked Cameraman **b** Tampered Cameraman, and **c** Actual tampered areas

The suggested extraction mechanism is employed to extract the watermark bits or authentication code. The detail watermark extraction procedure is further illustrated in Fig. 7. The algorithmic steps of this process are given in Algorithm 2.

Example 3 An illustration of the extraction process of the watermark is presented with reference to Example 2. The watermarked pixels are 157 and 118 and the secret key is $(101)_2$.

Solution The first watermarked pixel, $Pw = (157)_{10} = (1\,0\,\underline{0\,1\,1\,1\,0\,1})_2$ and adjusted pixel value is $Sw = W\_Component - 2$ in decimal (i.e. information bits from first six LSB position of the first watermarked pixel, Pw), so $Sw = (\underline{0\,1\,1\,1\,0\,1})_2 = (29)_{10}$. Similarly, for the second watermarked pixel, $Qw = (118)_{10} = (0\,1\,\underline{1\,1\,0\,1\,1\,0})_2$ find $Tw = W\_Component - 2 = (\underline{1\,1\,0\,1\,1\,0})_2 = 54$. Now, compute the difference value of Sw and Tw based on $diffw = |Sw - Tw| = |29 - 54| = 25$. The $diffw$ value belong to region R_4 in Fig. 2. The encrypted watermark, EW is extracted as follows:

$$EW = diffw - lower\_range_i = 25 - 24 = (1)_{10} = (0\,0\,1)_2.$$

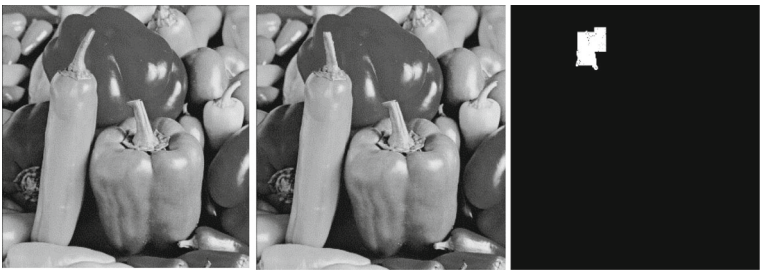
The original watermark, $W = EW \oplus K$.

Watermark bits, $W = (0\ 0\ 1)_2 \oplus (1\ 0\ 1)_2 = (1\ 0\ 0)_2$. Finally, extracted original watermark bits are $(100)_2$.

Input: Grayscale watermarked image (*WI*), Seed values of Logistic map
Output: Watermark bit streams

Begin

1. Read a grayscale watermarked image, *WI*.
2. Decompose *WI* into non-overlapping blocks of size *1×2*.
3. Read two consecutive pixels i.e. *Pw* and *Qw* from each block and convert each pixel into eight bits of binary representation.
4. Process each block by selecting *Sw=W_Component-2* in decimal from the first pixel, *Pw* and *Tw=W_Component-2* in decimal from the second pixel, *Qw*.
5. Compute the difference value $diffw = |Sw - Tw|$ and find the $lower\_range_i$ if the $diffw$ belong to R_i (Consider the table as shown in Fig 2).
6. Obtain the encrypted watermark, $EW = diffw - lower\_range_i$ and convert it in binary of 3 bits representation.
7. Based on the seed values of logistic map, find the secret key K using Eq.6 to Eq.9
8. Extract the watermark bits from the block as $W = EW \oplus K$
9. Process rest of the watermarked blocks using steps (3-8) for obtaining rest of watermark bits.



(a) (b) (c)

Fig. 17 **a** The watermarked Pepper **b** Tampered Pepper, and **c** Actual tampered areas

3.3 Tamper detection and localization region

The recipient received the seed values as secret keys for the Logistic-map from the sender side. Recipient generates 3 bits of the secret key sequence using the Logistic map. Then, the recipient computes three bits of the watermark (W) in a similar procedure as done by the sender side. Further, the recipient extracts the watermark bits from the watermarked image. These bits are checked by comparing the extracted watermark with the generated watermark at the receiver side. If the extracted watermark matches with the generated watermark, then the watermarked image has not tampered. Figure 8 briefly outline the tamper detection procedure, and the detailed procedure is further demonstrated in Fig. 9. Algorithm 3 describes the procedure for tamper detection procedure in proper order.

Example 4 The receiver side detects the extracted tampered watermark bits. Consider all parameters of *Example 2* and suppose the recipient received tampered watermarked pixels as 153 and 116 with secret key bits as $(101)_2$.

Solution The first tampered watermarked pixels $Pw_T = (153)_{10} = (1\,0\,\underline{0\,1\,1\,0\,0\,1})_2$ and adjusted pixel value is $Sw_T = W\_Component - 2$ in decimal (i.e. information bits from first six LSB position of the first tampered watermarked pixel, Pw_T), so $Sw_T = (\underline{0\,1\,1\,0\,0\,1})_2 = (25)_{10}$. Similarly, for the second tampered watermarked pixel,

(a) (b) (c)

Fig. 18 **a** The watermarked Barbara **b** Tampered Barbara, and **c** Actual tampered areas

(a) (b) (c)

Fig. 19 **a** The watermarked Boat **b** Tampered Boat, and **c** Actual tampered areas

$Qw_T = (116)_{10} = (0\,1\,\underline{1\,1\,0\,1\,0\,0}\,)_2$ find $Tw_T = W\_Component{-}2 = (\underline{1\,1\,0\,1\,0\,0}\,)_2 = 52$. Now, compute the difference value of Sw_T and Tw_T based on $diffw_T = |Sw_T - Tw_T| = |25 - 52| = 27$. The $diffw_T$ value belong to region R_4 in Fig. 2. The encrypted watermark, EW is extracted as follows: $EW = diffw_T - lower\_range_i = 27 - 24 = (3)_{10} = (011)_2$. The extracted watermark bits, $W = EW \oplus K = (011)_2 \oplus (101)_2 = (110)_2$. Later, from first tampered watermarked pixel Pw_T, x = *Component-1* (i.e. information bits from 1st and 2nd position MSB content of Pw_T,) $= (10)_2$. Similarly, for the second tampered

Input: Received watermarked/tampered image (*WI*)
Output: Tamper block identification
Begin

1. Read the watermarked/tampered image, *WI*.
2. Decompose the watermarked/tampered image into non-overlapping blocks of size *1×2* pixels.
3. Read two consecutive pixels i.e. *Pw* and *Qw* from each block and convert each pixel into eight bits of binary representation.
4. Process each block by selecting *x=Component-1* and *Sw=W_Component-2* in decimal from *Pw*. similarly for *Qw* compute *y=Component-1* and *Tw=W_Component-2* in decimal form.
5. Append *x* and *y*, i.e. *z=x||y* to get the data in 4-bit form as $z_{1\times4}$=[D_1 D_2 D_3 D_4]
6. Compute the Hamming code using the symmetric Hamming generator matrix (G): $\Upsilon = z_{1\times4} \times G_{4\times7}$ and the 3-bit parity is obtained from the first 3 bits LSB of the Hamming code
7. Consider the parity bits as watermark bits.
8. Find the extracted watermark bits using Algorithm 2 from the given watermarked/tampered image.
9. If (*watermark 3 bits ≠ extracted watermark 3 bits*)
   Image block is tampered
   Else
   Image block is not tampered.
   If End
10. Process rest of the watermarked blocks using steps (3-9) to locate whether the block is tampered or not.

End

Table 5 Experimental results of the proposed scheme in terms of *TP*, *TN*, *FP*, *FN*, *TPR*, *FPR*, *FNR*, *Accuracy (ACC)* and *NC*

Image Name (512 × 512)	TP	TN	FP	FN	FPR	FNR	TPR	Accuracy (ACC)	NC
Lena	3667	197,856	167	54	0.0008	0.0145	0.9855	0.9989	0.9937
Baboon	2402	199,083	20	123	0.0001	0.0487	0.9513	0.9993	0.9983
Goldhill	3176	197,106	34	7	0.0002	0.0022	0.9978	0.9998	0.9984
Crowd	3537	190,487	32	7	0.0002	0.0020	0.9980	0.9998	0.9991
Cameraman	4856	188,757	130	54	0.0007	0.0110	0.9890	0.9991	0.9984
Pepper	3114	198,563	64	26	0.0003	0.0083	0.9917	0.9996	0.9966
Barbara	3072	197,586	12	12	0.0001	0.0039	0.9961	0.9999	0.9988
Boat	3418	196,500	109	41	0.0006	0.0119	0.9881	0.9993	0.9934

watermarked pixel, Qw_T, $y = Component\text{-}1 = (01)_2$. Now, four bit data is $z = x \| y = (1\ 0\ 0\ 1)_2$

$$T = z_{1\times 4} \times G_{4\times 7} = [1 \quad 0 \quad 0 \quad 1] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = \left(1001\ \underline{100}\right)_2.$$

The watermark bits are $(100)_2$.

Hence, watermark bits ≠ extracted watermark bits. So, the block has been identified as tampered in the watermarked image.

4 Experimental results

In this section, the experimental results are demonstrated in two aspects. Firstly, we have presented the experimental results according to the various perceptual quality measures. Subsequently, we have demonstrated the results in order to evaluate the tamper detection capabilities of the proposed work. For depicting the performance of the proposed scheme, eight different types of grayscale cover images have been considered. These cover images are of size *512 × 512* which are shown in Fig. 10(a-

Table 6 Comparison in terms of *FPR* and *FNR* values

Image Name 512 × 512	Chang et al. [3]		Tong et al. [21]		Nazari et al. [11]		Sreenivas et al. [19] Method-A		Proposed Scheme	
	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR
Lena	0.0037	0.0037	0.0081	0.0106	0.0071	0.0065	0.0037	0.0036	0.0008	0.0145
Baboon	0.3947	0.5450	0.0528	0.0007	0.0018	0.1430	0.0002	0.0504	0.0001	0.0487
Goldhill	0.0012	0.0016	0.0033	0.0018	0.0041	0.0083	0.0036	0.0021	0.0002	0.0022
Crowd	0.0028	0.0028	0.0043	0.0098	0.0016	0.0122	0.0047	0.0041	0.0002	0.0020
Cameraman	0.0024	0.0025	0.0048	0.0109	0.0020	0.0029	0.0025	0.0027	0.0007	0.0110
Pepper	0.0056	0.0052	0.0190	0.0144	0.0132	0.0135	0.0032	0.0038	0.0003	0.0083
Barbara	0.0004	0.0062	0.0013	0.0046	0.0019	0.0132	0.0021	0.0016	0.0001	0.0039
Boat	0.0055	0.0058	0.0071	0.0062	0.0046	0.0058	0.0042	0.0041	0.0006	0.0119

Table 7 Comparison in terms of *Accuracy (ACC)* and *NC* values

Image Name (512 × 512)	Chang et al. [3]		Tong et al. [21]		Nazari et al. [11]		Sreenivas et al. [19] Method-A		Proposed Scheme	
	ACC	NC	ACC	NC	ACC	NC	ACC	NC	ACC	NC
Lena	0.9963	0.9874	0.9909	0.9610	0.9930	0.9512	0.9964	0.9876	0.9989	0.9937
Baboon	0.9995	0.9305	0.9716	0.9916	0.9493	0.9846	0.9993	0.9963	0.9993	0.9983
Goldhill	0.9986	0.9975	0.9974	0.9952	0.9948	0.9819	0.9972	0.9943	0.9998	0.9984
Crowd	0.9972	0.9926	0.9942	0.9852	0.9949	0.9977	0.9956	0.9899	0.9998	0.9991
Cameraman	0.9976	0.9935	0.9972	0.9944	0.9976	0.9920	0.9974	0.9934	0.9991	0.9984
Pepper	0.9946	0.9874	0.9825	0.9564	0.9867	0.9209	0.9965	0.9905	0.9996	0.9966
Barbara	0.9967	0.9991	0.9955	0.9969	0.9947	0.9927	0.9982	0.9957	0.9999	0.9988
Boat	0.9944	0.9886	0.9933	0.9893	0.9949	0.9703	0.9958	0.9916	0.9993	0.9934

h). The resultant images after watermark embedding are exhibited in Fig. 11(a-h) and these results reveal that the visual quality of watermarked images meet the required standards since the watermarked images reveal close similarity with the original cover images under the observation of human visual perception. The perceptual quality of the watermarked images are further measured by some parameters like peak signal to noise ratio (*PSNR*), structural similarity index (*SSIM*) and image fidelity (*IF*). The *PSNR* is computed based on Eq. 10.

$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{MSE}\right) dB \tag{10}$$

where MSE stands for mean squared error and it is computed using given Eq.11:

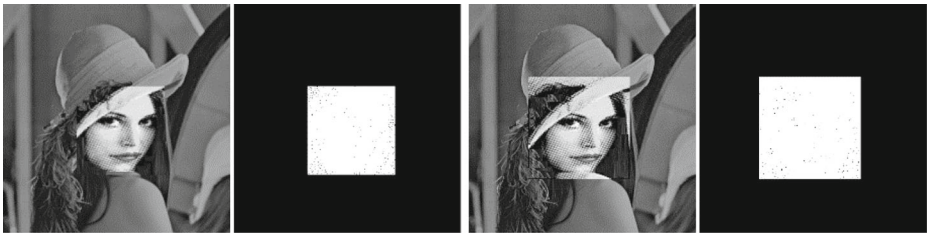
$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I_c(i,j) - I_w(i,j)]^2 \tag{11}$$

where I_c is a cover image, I_w is a watermarked image, M represents the width and N represents the height of I_c as well as I_w.

High value of peak signal to noise ratio implies higher similarities between a watermarked image and corresponding original cover image. Table 1 depicts that proposed scheme retains considerably high *PSNR* values even after embedding high payload into the cover image. In all cases the *PSNR* values are greater than 40 *dB*.

(a) (b) (c) (d)

Fig. 20 **a** 5% tampered Lena **b** 5% actual tampered areas, **c** 10% tampered Lena **d** 10% actual tampered areas



(a) (b) (c) (d)

Fig. 21 **a** 15% tampered Lena **b** 15% actual tampered areas, **c** 20% tampered Lena **d** 20% actual areas

This shows that the watermarked images generated by the proposed procedure, are very close to the original cover image.

The proposed scheme is evaluated further by other quality parameters like image fidelity (*IF*) and structural similarity (*SSIM*) index. The typical *SSIM* and *IF* values ranges from 0 to 1. The *SSIM* values are calculated using following Eq. 12.

$$SSIM(I_c,\ I_w) = \frac{(2M_{Ic}M_{Iw} + C_1)(2\mathrm{cov}(I_w) + C_2)}{\left(M^2{}_{Ic} + M^2{}_{Iw} + C_1\right)(\sigma^2{}_{Ic} + \sigma^2{}_{Iw} + C_2)} \tag{12}$$

where, $\begin{cases} C_1 = (k_1 D)^2 & k_1 = 0.01 \\ C_2 = (k_2 D)^2 & k_2 = 0.03 \end{cases}$ and.

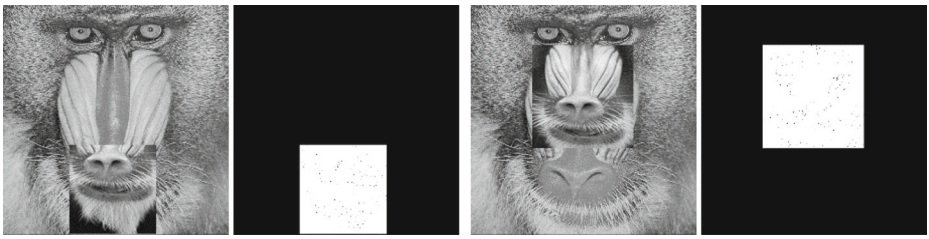
M_{Ic} and M_{Iw} are the mean of the original cover image and watermarked image respectively, $\sigma^2{}_{Ic}$ and $\sigma^2{}_{Iw}$ are the variance of the original cover image and watermarked image respectively, *cov* is the covariance of the watermarked image, C_1 and C_2 are variables for finding constant division with weak dominator and $D = 255$ which is the self-motivated range of pixel values.

The image fidelity (*IF*) is another quality measure which can be computed using Eq. 13.

$$IF = 1 - \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left[I_c(i,j) - I_w\left(i,j\right)\right]^2}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left[I_c(i,j)\right]^2} \tag{13}$$

(a) (b) (c) (d)

Fig. 22 **a** 5% tampered Baboon **b** 5% actual tampered areas, **c** 10% tampered Baboon **d** 10% actual tampered areas



(a) (b) (c) (d)

Fig. 23 **a** 15% tampered Baboon **b** 15% actual tampered areas, **c** 20% tampered Baboon **d** 20% actual tampered areas

where I_c is a cover image, I_w is a watermarked image, M represents the width and N represents the height of I_c as well as I_w.

Table 1 also shows the *SSIM* and *IF* values obtained on various test images. The proposed watermarking procedure has achieved very high *SSIM* and *IF* values which are close to 1. This represents that the watermarked images obtained by the proposed scheme are of notably high quality. So this procedure is suitable in meeting the indispensable criteria required for efficient fragile watermarking in terms of holding high visual similarity of watermarked images with the original cover images. The proposed scheme is also compared with some other related works in terms of perceptual quality of the watermarked images. Tables 2, 3, and 4 show a comparative study of the proposed scheme with other schemes proposed by Chang et al. [3], Tong et al. [21], Nazari et al. [11] and Sreenivas et al. [19] Method-A. The analysis shows that the proposed scheme produces higher quality results as compared to the other works as mentioned in Tables 2, 3, and 4 respectively.

4.1 Tamper detection evaluation

We have carried out several intentional attacks on the watermarked images to validate the effectiveness of the proposed scheme in terms of tamper detection. In the watermarked "Lena" image, the eyes have been altered by adding eyeglasses in Fig. 12b. Figure 12c shows the actually tampered areas for the "Lena" image. The seven other images used in the experiment have been altered in some other way. In the watermarked "Baboon" image, the face of the Baboon has been altered (Fig. 13b); in the "Goldhill" image, a window in the background has been concealed (Fig. 14b); in the "Crowd" image, a boy's face has been altered

(a) (b) (c) (d)

Fig. 24 **a** 5% tampered Goldhill **b** 5% actual tampered areas, **c** 10% tampered Goldhill **d** 10% actual tampered areas

(a) (b) (c) (d)

Fig. 25 **a** 15% tampered Goldhill **b** 15% actual tampered areas, **c** 20% tampered Goldhill **d** 20% actual tampered areas

(Fig. 15b); in the "Cameraman" image, the tower in the background has been added (Fig. 16b); in the watermarked "Pepper" image the direction of one of the vegetable has been altered (Fig. 17b); in the "Barbara" image the bracelet in the wrists have been added (Fig. 18b), and in the "Boat" image the tower in the boat background has been added (Fig. 19b). In all the cases, the proposed scheme is able to identify the tampered region successfully. The effectiveness of the scheme is estimated by several quantitative measures like false positive rate (*FPR*), false negative rate (*FNR*) and true positive rate (*TPR*). Proposed watermarking scheme obtained very less value of *FPR* and *FNR*. False positive rate (*FPR*), false negative rate (*FNR*) and true positive rate (*TPR*) are used to evaluate the tamper detection correctness using Eq. 14 to Eq. 16.

$$FPR = \frac{FP}{(FP + TN)} \tag{14}$$

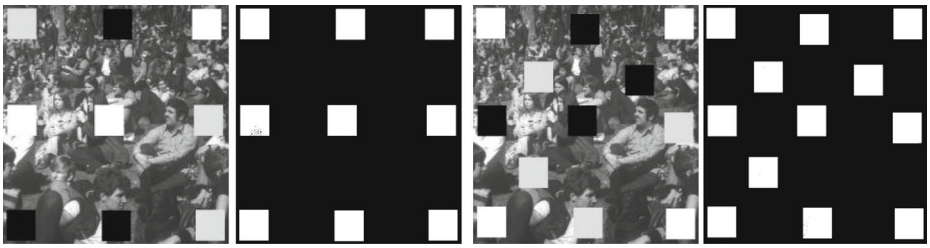
$$FNR = \frac{FN}{(FN + TP)} \tag{15}$$

$$TPR = \frac{TP}{(TP + FN)} \tag{16}$$

where, *FP* is the number of false positive pixels, *FN* is the number of false negative pixels, *TP* is the number of true positive pixels, and *TN* is the number of true negative pixels. Proposed fragile watermarking scheme ensures low *FPR* and *FNR*. Table 5 shows quite good results in terms of *TP*, *TN*, *FP*, *FN*, *TPR*, *FPR* and *FNR*. *Accuracy (ACC)* is another parameter for tamper detection evaluation in watermarked images. The *Accuracy (ACC)* of tamper detection is derived using Eq. 17:

(a) (b) (c) (d)

Fig. 26 **a** 5% tampered Crowd **b** 5% actual tampered areas, **c** 10%tampered Crowd **d** 10% actual tampered areas



(a) (b) (c) (d)

Fig. 27 **a** 15% tampered Crowd **b** 15% actual tampered areas, **c** 20%tampered Crowd **d** 20% actual tampered areas

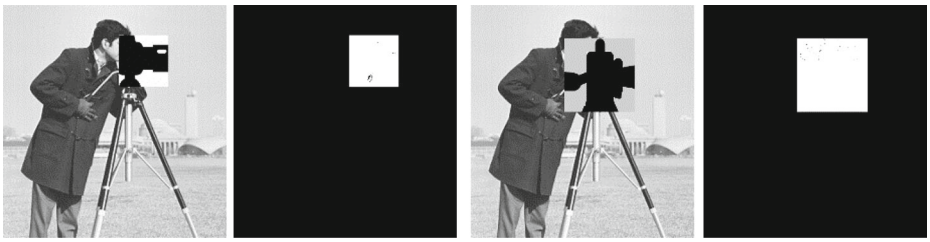
$$Accuracy\ (ACC) = \frac{TN + TP}{(FP + TN + FN + TP)} \tag{17}$$

where, *TP* is the number of tampered pixels stated tampered, *FP* is number of non-tampered pixels declared tampered, *TN* is number of non-tampered pixels declared non-tampered and *FN* is number of tampered pixels declared non-tampered. *Accuracy* value also ranges from 0 to 1. Normalized cross-correlation (*NC*) is another important quality measure for watermarking schemes which is used for calculating the reliability of extracted watermark bits. The value of *NC* must is close to 1 for retaining high quality watermark. The *NC* value is calculated based on Eq.18:

$$NC(OW, EW) = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[OW(i,j) - mean\ (OW)] \times [EW(i,j) - mean\ (EW)]}{\sqrt{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[OW(i,j) - mean\ (OW)]^2} \times \sqrt{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[EW(i,j) - mean\ (EW)]^2}} \tag{18}$$

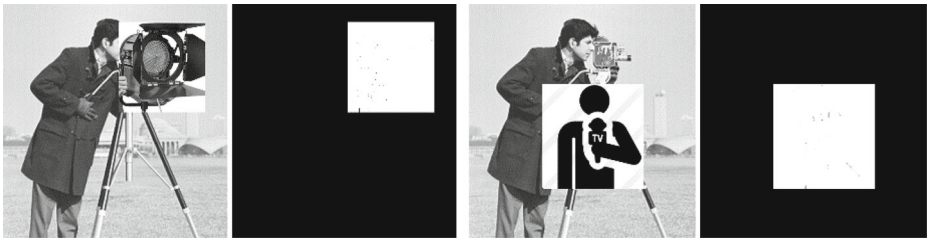
where, *M* and *N* represents the width and height of images, *OW* is an original watermark, *EW* is an extracted watermark, *mean*(*OW*) is a mean of original watermark and *mean*(*EW*) is the mean of the extracted watermark.

Table 5 also demonstrates the *Accuracy* and normalized cross-correlation values attained by the proposed tamper detection process. The proposed scheme attains both *Accuracy* and *NC* values very close to 1 which validates the effectiveness of the watermark extraction, and the tamper detection process. Tables 6 and 7 shows a comparative analysis of the proposed scheme with the schemes proposed by Chang et al. [3], Tong et al. [21], Nazari et al. [11], and Sreenivas et al. [19] Method-A on the



(a) (b) (c) (d)

Fig. 28 **a** 5% tampered Cameraman **b** 5% actual tampered areas, **c** 10%tampered Cameraman **d** 10% actual tampered areas



(a) (b) (c) (d)

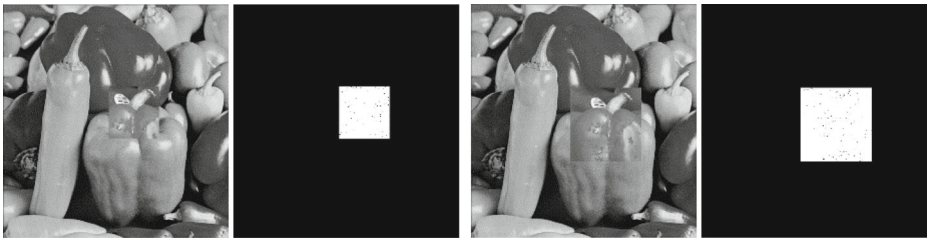
Fig. 29 **a** 15% tampered Cameraman **b** 15% actual tampered areas, **c** 20%tampered Cameraman **d** 20% actual tampered areas

basis of various parameters like true positive rate, false positive rate, tamper detection accuracy, and normalized cross-correlation. The results obtained by the proposed tamper detection procedure is found to be comparable with the other mentioned schemes as the proposed procedure achieves considerably low *FPR* and *FNR* values with very high *Accuracy* and *NC* values.

4.2 Percentage of tampering ratio

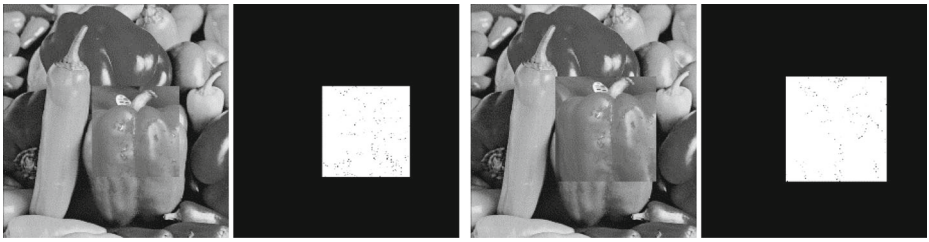
Percentage of tampering ratio is another parameter to estimate the performance of any fragile watermarking scheme. This experiment evaluates the efficiency of the watermarking scheme with respect to various levels of tampering.

The percentage of tampering ratio is computed as follows:



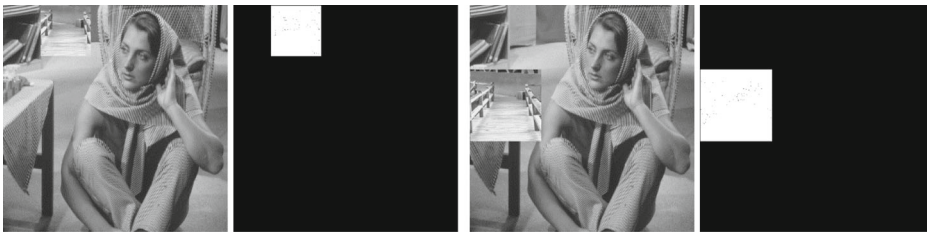
(a) (b) (c) (d)

Fig. 30 **a** 5% tampered Pepper **b** 5% actual tampered areas, **c** 10%tampered Pepper **d** 10% actual tampered areas



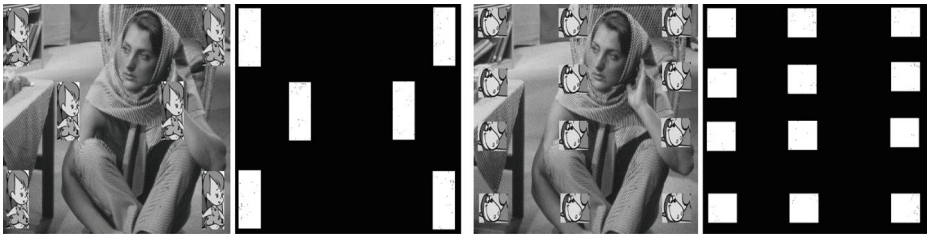
(a) (b) (c) (d)

Fig. 31 **a** 15% tampered Pepper **b** 15% actual tampered areas, **c** 20%tampered Pepper **d** 20% actual tampered areas



(a) (b) (c) (d)

Fig. 32 **a** 5% tampered Barbara **b** 5% actual tampered areas, **c** 10%tampered Barbara **d** 10% actual tampered areas



(a) (b) (c) (d)

Fig. 33 **a** 15% tampered Barbara **b** 15% actual tampered areas, **c** 20% tampered Barbara **d** 20% actual tampered areas

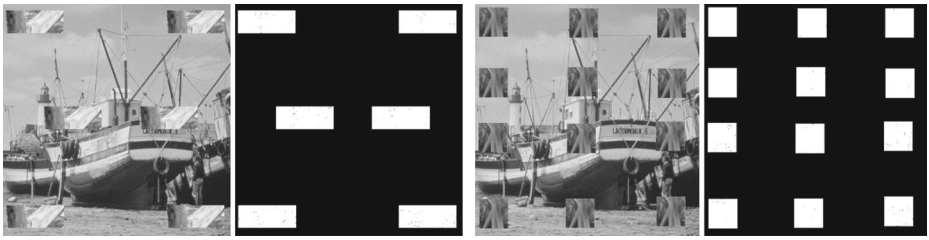
$$\text{The tampering ratio}\% = \frac{Ntb \times 100}{Nb}$$

where, *Nb* represents the number of blocks in the test image and *Ntb* implies the number of tampered blocks found after image manipulation attacks.

We have conducted the experiment with various percentages of tampering ratio as *5%, 10%, 15%,* and *20%* values and it is found that the proposed scheme has survived against these much distortions. Figures 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34 and 35 show the visual results with various percentages of tampering ratio in different images, where all the results depict that the tampered areas have been detected precisely by the proposed procedure. In most of the cases, the proposed scheme has retained acceptable evaluation values in terms of *FPR, FNR, Accuracy,* and *NC* as shown in Table 8. In addition, the proposed procedure has been compared with some related works in Tables 9 and 10, where it is observed that in almost all the instances it retains

(a) (b) (c) (d)

Fig. 34 **a** 15% tampered Boat **b** 15% actual tampered areas, **c** 20% tampered Barbara **d** 20% actual tampered areas



(a) (b) (c) (d)

Fig. 35 **a** 5% tampered Boat **b** 5% actual tampered areas, **c** 10% tampered Boat **d** 10% actual tampered areas

comparable results. Based on the presented experimental outcomes, it is found that the proposed fragile watermarking scheme is highly suitable for detecting tampered regions accurately. In addition, the scheme also enhances the level of security due to consideration of encrypted watermark or authentication code.

Table 8 Experiment results of the proposed scheme in terms of various percentage of tampering ratio

Image Name (512 × 512)	Proposed Method									
	Tampered Rate (%)	TP	TN	FP	FN	FPR	FNR	TPR	Accuracy	NC
Lena	5	3596	197,023	300	104	0.0015	0.0281	0.9719	0.9980	0.9826
	10	3435	195,797	606	196	0.0031	0.0540	0.9460	0.9960	0.9679
	15	3301	194,275	894	299	0.0046	0.0831	0.9169	0.9940	0.9504
	20	3156	192,989	1134	406	0.0058	0.1140	0.8860	0.9922	0.9335
Baboon	5	2389	197,816	287	90	0.0014	0.0363	0.9637	0.9981	0.9862
	10	2260	196,409	706	177	0.0036	0.0726	0.9274	0.9956	0.9777
	15	2152	195,059	912	245	0.0047	0.1022	0.8978	0.9942	0.9584
	20	2004	193,347	1295	319	0.0067	0.1373	0.8627	0.9918	0.9412
Gold-hill	5	3070	196,031	476	91	0.0024	0.0288	0.9712	0.9972	0.9927
	10	2969	194,820	905	168	0.0046	0.0536	0.9464	0.9946	0.9837
	15	2812	193,494	1336	258	0.0069	0.0840	0.9160	0.9919	0.9723
	20	2690	192,352	1701	364	0.0088	0.1192	0.8808	0.9895	0.9637
Crowd	5	3405	189,222	324	98	0.0017	0.0280	0.9720	0.9978	0.9872
	10	3250	187,965	667	184	0.0035	0.0536	0.9464	0.9956	0.9762
	15	3060	189,483	639	457	0.0034	0.1299	0.8701	0.9943	0.9999
	20	2864	188,827	877	607	0.0046	0.1749	0.8251	0.9923	0.9995
Cameraman	5	4716	188,780	154	208	0.0008	0.0422	0.9578	0.9981	0.9988
	10	4495	189,013	22	442	0.0001	0.0895	0.9105	0.9976	0.9992
	15	4126	187,040	292	645	0.0016	0.1352	0.8648	0.9951	0.9663
	20	4022	188,442	126	878	0.0007	0.1792	0.8208	0.9948	0.9896
Pepper	5	3036	197,530	371	79	0.0019	0.0254	0.9746	0.9978	0.9892
	10	2938	196,225	883	151	0.0045	0.0489	0.9511	0.9948	0.9826
	15	2817	195,077	1379	252	0.0070	0.0821	0.9179	0.9918	0.9774
	20	2686	193,760	1763	345	0.0090	0.1138	0.8862	0.9894	0.9669
Barbara	5	2911	196,378	469	126	0.0024	0.0415	0.9585	0.9970	0.9921
	10	2810	195,132	934	206	0.0048	0.0683	0.9317	0.9943	0.9839
	15	2638	193,478	1007	283	0.0052	0.0969	0.9031	0.9935	0.9549
	20	2517	192,797	1261	376	0.0065	0.1300	0.8700	0.9917	0.9500
Boat	5	3340	195,595	280	91	0.0014	0.0265	0.9735	0.9981	0.9820
	10	3207	194,405	477	190	0.0024	0.0559	0.9441	0.9966	0.9686
	15	3034	193,195	683	300	0.0035	0.0900	0.9100	0.9950	0.9535
	20	2846	191,134	1174	395	0.0061	0.1219	0.8781	0.9920	0.9312

Table 9 Comparison in terms of *FPR* and *FNR* values under various percentage of tampering ratio

Image Name (512 × 512)	Tampered Rate (%)	Chang et al. [3]		Tong et al. [21]		Nazari et al. [11]		Sreenivas et al. [19] Method-A		Proposed Method	
		FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR
Lena	5	0.0058	0.0051	0.0153	0.0149	0.0121	0.0081	0.0062	0.0058	0.0015	0.0281
	10	0.0153	0.0197	0.0251	0.0382	0.0279	0.0267	0.0133	0.0108	0.0031	0.0540
	15	0.0237	0.0338	0.0403	0.0589	0.0442	0.0492	0.0227	0.0177	0.0046	0.0831
	20	0.0351	0.0482	0.0528	0.0843	0.0615	0.0633	0.0300	0.0229	0.0058	0.1140
Baboon	5	0.0070	0.0057	0.0133	0.0133	0.0120	0.0155	0.0066	0.0036	0.0014	0.0363
	10	0.0132	0.0127	0.0296	0.0280	0.0235	0.0341	0.0145	0.0105	0.0036	0.0726
	15	0.0206	0.0178	0.0432	0.0428	0.0366	0.0596	0.0219	0.0168	0.0047	0.1022
	20	0.0287	0.0237	0.0574	0.0569	0.0497	0.0840	0.0329	0.0298	0.0067	0.1373
Gold-hill	5	0.0064	0.0051	0.0140	0.0148	0.0110	0.0185	0.0057	0.0060	0.0024	0.0288
	10	0.0146	0.0111	0.0326	0.0253	0.0284	0.0236	0.0148	0.0115	0.0046	0.0536
	15	0.0225	0.0172	0.0497	0.0425	0.0461	0.0333	0.0218	0.0168	0.0069	0.0840
	20	0.0292	0.0256	0.0551	0.0675	0.0600	0.0500	0.0289	0.0247	0.0088	0.1192
Crowd	5	0.0062	0.0056	0.0141	0.0163	0.0144	0.0139	0.0062	0.0067	0.0017	0.0280
	10	0.0158	0.0115	0.0283	0.0299	0.0291	0.0243	0.0140	0.0133	0.0035	0.0536
	15	0.0225	0.0172	0.0428	0.0506	0.0453	0.0396	0.0220	0.0214	0.0034	0.1299
	20	0.0285	0.0261	0.0551	0.0675	0.0599	0.0519	0.0296	0.0280	0.0046	0.1749
Cameraman	5	0.0069	0.0049	0.0142	0.0095	0.0163	0.0120	0.0070	0.0058	0.0008	0.0422
	10	0.0144	0.0106	0.0357	0.0170	0.0373	0.0224	0.0137	0.0118	0.0001	0.0895
	15	0.0256	0.0164	0.0412	0.0230	0.0588	0.0348	0.0228	0.0182	0.0016	0.1352
	20	0.0318	0.0239	0.0602	0.0393	0.0588	0.0348	0.0294	0.0246	0.0007	0.1792
Pepper	5	0.0068	0.0064	0.0125	0.0192	0.0115	0.0152	0.0052	0.0061	0.0019	0.0254
	10	0.0135	0.0139	0.0263	0.0322	0.0244	0.0288	0.0118	0.0151	0.0045	0.0489
	15	0.0209	0.0227	0.0387	0.0512	0.0442	0.0422	0.0172	0.0205	0.0070	0.0821
	20	0.0282	0.0310	0.0477	0.0826	0.0618	0.0620	0.0288	0.0244	0.0090	0.1138
Barbara	5	0.0069	0.0064	0.0114	0.0109	0.0103	0.0233	0.0068	0.0059	0.0024	0.0415
	10	0.0145	0.0109	0.0262	0.0205	0.0261	0.0332	0.0144	0.0121	0.0048	0.0683
	15	0.0217	0.0172	0.0422	0.0321	0.0419	0.0566	0.0209	0.0183	0.0052	0.0969
	20	0.0315	0.0239	0.0601	0.0434	0.0554	0.0756	0.0296	0.0267	0.0065	0.1300
Boat	5	0.0064	0.0056	0.0135	0.0148	0.0082	0.0105	0.0077	0.0046	0.0014	0.0265
	10	0.0146	0.0103	0.0257	0.0280	0.0247	0.0257	0.0145	0.0108	0.0024	0.0559
	15	0.0206	0.0177	0.0393	0.0395	0.0374	0.0442	0.0222	0.0184	0.0035	0.0900
	20	0.0285	0.0228	0.0593	0.0526	0.0487	0.0626	0.0306	0.0269	0.0061	0.1219

Table 10 Comparison in terms of *ACC* and *NC* values under various percentage of tampering ratio

Image Name (512 × 512)	Tam-ered Rate (%)	Chang et al. [3]		Tong et al. [21]		Nazari et al. [11]		Sreenivas et al. [19] Method-A		Proposed Method	
		ACC	NC	ACC	NC	ACC	NC	ACC	NC	ACC	NC
Lena	5	0.9945	0.9782	0.9849	0.9559	0.9887	0.9317	0.9940	0.9837	0.9980	0.9826
	10	0.9826	0.9629	0.9696	0.9141	0.9723	0.8665	0.9880	0.9677	0.9960	0.9679
	15	0.9715	0.9588	0.9522	0.8845	0.9549	0.8151	0.9798	0.9528	0.9940	0.9504
	20	0.9587	0.9451	0.9344	0.8478	0.9381	0.7605	0.9736	0.9348	0.9922	0.9335
Baboon	5	0.9937	0.9839	0.9867	0.9769	0.9868	0.9409	0.9949	0.9821	0.9981	0.9862
	10	0.9871	0.9704	0.9711	0.9531	0.9729	0.8857	0.9875	0.9691	0.9956	0.9777
	15	0.9808	0.9532	0.9570	0.9300	0.9557	0.8350	0.9806	0.9525	0.9942	0.9584
	20	0.9738	0.9376	0.9428	0.9083	0.9389	0.7912	0.9687	0.9365	0.9918	0.9412
Gold-hill	5	0.9942	0.9833	0.9857	0.9678	0.9870	0.9420	0.9942	0.9833	0.9972	0.9927
	10	0.9871	0.9661	0.9707	0.9379	0.9729	0.8827	0.9869	0.9677	0.9946	0.9837
	15	0.9801	0.9478	0.9536	0.9097	0.9574	0.8328	0.9807	0.9536	0.9919	0.9723
	20	0.9726	0.9348	0.9297	0.8674	0.9427	0.7897	0.9733	0.9366	0.9895	0.9637
Crowd	5	0.9940	0.9819	0.9853	0.9596	0.9857	0.9271	0.9936	0.9853	0.9978	0.9872
	10	0.9863	0.9643	0.9713	0.9210	0.9725	0.8617	0.9864	0.9705	0.9956	0.9762
	15	0.9800	0.9478	0.9551	0.8901	0.9566	0.8018	0.9783	0.9563	0.9943	0.9999
	20	0.9726	0.9332	0.9416	0.8574	0.9427	0.7532	0.9712	0.9410	0.9923	0.9995
Cameraman	5	0.9941	0.9816	0.9872	0.9619	0.9855	0.9391	0.9936	0.9822	0.9981	0.9988
	10	0.9875	0.9685	0.9698	0.9266	0.9691	0.8843	0.9873	0.9668	0.9976	0.9992
	15	0.9791	0.9499	0.9642	0.8776	0.9515	0.8393	0.9795	0.9490	0.9951	0.9663
	20	0.9722	0.9347	0.9460	0.8603	0.9515	0.8393	0.9730	0.9358	0.9948	0.9896
Pepper	5	0.9934	0.9845	0.9853	0.9667	0.9874	0.9182	0.9944	0.9854	0.9978	0.9892
	10	0.9863	0.9687	0.9717	0.9348	0.9742	0.8475	0.9865	0.9716	0.9948	0.9826
	15	0.9782	0.9530	0.9573	0.9029	0.9564	0.7715	0.9812	0.9559	0.9918	0.9774
	20	0.9704	0.9379	0.9413	0.8759	0.9381	0.7190	0.9734	0.9340	0.9894	0.9669
Barbara	5	0.9934	0.9831	0.9888	0.9664	0.9859	0.9331	0.9937	0.9837	0.9970	0.9921
	10	0.9873	0.9668	0.9762	0.9324	0.9718	0.8677	0.9867	0.9677	0.9943	0.9839
	15	0.9805	0.9497	0.9622	0.8983	0.9538	0.8091	0.9804	0.9523	0.9935	0.9549
	20	0.9722	0.9340	0.9471	0.8677	0.9387	0.7603	0.9718	0.9368	0.9917	0.9500
Boat	5	0.9940	0.9842	0.9860	0.9688	0.9908	0.9322	0.9939	0.9812	0.9981	0.9820
	10	0.9876	0.9672	0.9735	0.9408	0.9748	0.8457	0.9874	0.9646	0.9966	0.9686
	15	0.9809	0.9526	0.9606	0.9206	0.9596	0.7870	0.9797	0.9504	0.9950	0.9535
	20	0.9744	0.9373	0.9431	0.8954	0.9451	0.7339	0.9713	0.9360	0.9920	0.9312

5 Conclusions

A secure fragile watermarking scheme is suggested in this paper. The watermark or authentication code is generated using Hamming code from the most significant bits of each pixel. For maintaining the high level of security, the watermark bits are further encrypted by secret binary bits obtained from the Logistic map. The encrypted bits are embedded into the cover image using the block-level pixel adjustment process. Suggested watermark embedding procedure causes less distortion since it is an indirect data hiding process. The proposed scheme is evaluated on various grayscale images and expected outcomes have been achieved in both perceptual observation as well as computed evaluation parameters like *PSNR, IF, Accuracy, NC, TPR, FPR,* and *FNR*. The proposed work has been tested with various percentage of tampering ratios, and the results show that the generated outcomes are of desirable quality. As per the experimental results, it can be stated that the proposed scheme is an effective and secure fragile image watermarking scheme for detecting tampered regions in digital images.

References

1. Bravo-Solorio S, Calderon F, Li C-T, Nandi AK (2018) Fast fragile watermark embedding and iterative mechanism with high self-restoration performance. Digital Signal Processing 73:83–93
2. Chana C-S, Chang C-C (2007) An efficient image authentication method based on Hamming code. Pattern Recogn 40:681–690
3. Chang CC, Chen KN, Lee CF, Liu LJ (2011) A secure fragile watermarking scheme based on chaos-and-hamming code. J Syst Softw 84(9):1462–1470
4. Chang Y-F, Tail W-L (2013) A block-based watermarking scheme for image tamper detection and self-recovery. Opto-Electronics Review 21(2):182–190
5. Chen F, He H, Tai H-M, Wang H (2012) Chaos-based self-embedding fragile watermarking with flexible watermark payload. Multimed Tools Appl 72(1):41–56
6. He, H, Zhang, J, Chen, F (2007) Block-wise fragile watermarking scheme based on scramble encryption. In Bio-Inspired Computing: Theories and Applications, 2007. BIC-TA 2007. Second International Conference on (pp. 216–220). IEEE
7. Hsu C-S, Tu S-F (2016) Image tamper detection and recovery using adaptive embedding rules. Measurement 88:287–296
8. Kaur Y, Ranade SK (2017) Image authentication and tamper detection using fragile watermarking in spatial domain. International Journal of Advanced Research in Computer Engineering and Technology (IJARCET) 6(7):1058–1063
9. Lo C-C, Hu Y-C (2014) A novel reversible image authentication scheme for digital images. Signal Process 98:174–118
10. Lu C-S (2005) Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property. Idea Group Publishing
11. Nazari M, Sharif S (2017) An improved method for digital image fragile watermarking based on chaotic maps. Multimed Tools Appl 76(15):16107–16123
12. Peng Y, Niu X, Fu L, Yin Z (2018) Image authentication scheme based on reversible fragile watermarking with two images. Journal of Information Security and Applications 40(2018):236–246
13. Qi X, Xin X (2011) A quantization-based semi-fragile watermarking scheme for image content authentication. J Vis Commun Image Represent 22(2):187–200
14. Qi X, Xin X (2015) A singular-value-based semi-fragile watermarking scheme for image content authentication with tamper localization. J Vis Commun Image Represent 23(2015):312–327
15. Qin C, Ji P, Wang J, Chang C-C (2017) Fragile image watermarking scheme based on VQ index sharing and self-embedding. Multimed Tools Appl 76(2):2267–2287
16. Qin C, Ji P, Zhang X, Dong J, Wang J (2017) Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. Signal Process 138:280–293
17. Rawat S, Raman B (2011) A chaotic system based fragile watermarking scheme for image tamper detection. International Journal of Electronics and Communications (AEU) 65(10):840–847

18. Singh P, Chadha RS (2013) A Survey of Digital Watermarking Techniques, Applications and Attacks. International Journal of Engineering and Innovative Technology 2(9):165–175
19. Sreenivas K, Kamakshiprasad V (2017) Improved image tamper localisation using chaotic maps and self-recovery. J Vis Commun Image Represent 49:164–176
20. Suthaharan S (2010, 2010) Logistic Map-Based Fragile Watermarking for Pixel Level Tamper Detection and Resistance. EURASIP J Inf Secur:7
21. Tong X, Liu Y, Zhang M, Chen Y (2013) A novel chaos-based fragile watermarking for image tampering detection and self-recovery. Signal Process Image Commun 28(3):301–308
22. Trivedy S, Pal AK (2017) A Logistic Map-Based Fragile Watermarking Scheme of Digital Images with Tamper Detection. Iranian Journal of Science and Technology, Transactions of Electrical Engineering 41(2): 1–11
23. William S (2007) Cryptography and Network Security: Principles and Practices. 4th Ed. India; Pearson Education
24. Wu D-C, Tsai W-H (2003) A Steganographic Method for Images by Pixel Value Differencing. Pattern Recogn Lett 24:1613–1626
25. Xiao D, Shih FY (2012) An improved hierarchical fragile watermarking scheme using chaotic sequence sorting and subblock post- processing. Opt Commun 285(10–11):2596–2606
26. Yin Z, Niu X, Zhou Z, Tang J, Luo B (2016) Improved reversible image authentication scheme. Cogn Comput 8(5):890–899

Shiv Prasad received B.Sc. Degree from P.P.N. College (CSJMU) Kanpur, India, B.Ed. Degree from I.I.E.M. Hapur, India and M.C.A Degree from Kamla Nehru Institute of Technology (KNIT) Sultanpur, India. Currently, he is doing Ph.D. Degree from the Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines) Dhanbad, India. His research interest includes Steganography, Watermarking, Image security, and data hiding using Error-correcting code.

Arup Kumar Pal is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines) Dhanbad, India. He did his Ph.D. Degree in Computer Science and Engineering from Indian School of Mines, Dhanbad in 2011. He has around seven years of teaching and research experiences, and contributed a number of research papers in several Journals and Conference proceedings of National and International reputes. His main research interest includes Vector quantization, Image compression, Image Cryptosystem, Steganography, Watermarking and CBIR.