Multimedia Systems (2021) 27:531–545
https://doi.org/10.1007/s00530-021-00751-3

REGULAR PAPER

A novel pixel-wise authentication-based self-embedding fragile watermarking method

Ertugrul Gul[1,2] · **Serkan Ozturk**[2]

Received: 30 May 2020 / Accepted: 4 January 2021 / Published online: 5 February 2021

Abstract
Self-embedding fragile watermarking algorithms can perform detection of the manipulated areas as well as recovery of these detected areas. Most of the self-embedding fragile watermarking algorithms are performed block-wise authentication approach. However, entire block is marked as tampered in case one of the pixels in the block is detected as manipulated. This situation decreases the accuracy rate of the authentication process especially against pixel-based attacks such as salt and paper noise adding. Therefore, we present a novel pixel-wise authentication-based self-embedding fragile watermarking method for manipulation detection and recovery. In this proposed method, reference image is divided into four main blocks. Then, each main block is subdivided into 2×4 or 4×2 sized blocks according to block type determined using recovery quality. Recovery bits of each main block generated from sub-blocks are spreaded into the two main blocks in the other half of the image. For each pixel, two authentication bits are generated using the six most significant bits of the pixel with two pixel position bits and then embedded into the first and second least significant bits of the pixel. In experimental results, pixel-based attacks are applied to the images to demonstrate the success of the presented method. Also, performance of the presented method has been evaluated by applying different size of cropping attacks to the watermarked images. Experimental results show that the proposed method satisfactory detect and recover the manipulated areas.

Keywords Self-recovery · Self-embedding · Tamper detection · Pixel-wise

1 Introduction

Due to the increased transmission and reception of image data over the Internet, the image authentication has become an important issue for image applications in recent years [30–32]. To integrity and authentication of images, several digital signature-based schemes [6, 27] have been proposed. The digital signature-based schemes authenticate the image by comparison of the attached signature with the regenerated signature. The digital signature information can be either an image hashing or a character [25]. The major drawback of these methods is that they cannot localize the tampered areas, even though they can correctly detect that the image is altered [24]. Also, additional storage space is required to attach digital signature [25]. Therefore, to overcome these problems, many fragile watermarking-based schemes [4, 5, 7, 12, 15, 34] have been proposed for image authentication and tamper detection. Fragile watermarking methods embed an invisible fragile watermark into an image to detect slightest modifications and to localize the modificated areas.

Earlier fragile watermarking algorithms have only detected the modificated areas. However, detection of modificated areas is not sufficient for some applications such as image forensic analysis. In some cases, information about the original state of tampered areas is also required. Therefore, to overcome this challenge, several self-embedding watermarking algorithms [2, 8, 11, 13, 14, 22] have been proposed to recover the detected modificated areas. In self-embedding fragile watermarking methods, authentication and recovery bits are generated from the image itself. Then,

Communicated by L. Zhou.

✉ Ertugrul Gul
ertugrulgul@erciyes.edu.tr

Serkan Ozturk
serkan@erciyes.edu.tr

[1] Computer Engineering Department, Nigde Omer Halisdemir University, 51240 Nigde, Turkey

[2] Computer Engineering Department, Erciyes University, 38039 Kayseri, Turkey

these bits are embedded into the image as a watermark for detection and recovery of the modificated areas.

Self-embedding fragile watermarking algorithms can be divided into two categories in terms of recovery bits embedding approach: block mapping [1, 11, 22, 29] and bit mapping [3, 9, 16, 17]. In block mapping-based algorithms, recovery bits of the blocks are embedded into the mapped blocks by using block mapping. However, this algorithms cannot recover the tampered block when both the block and mapped block are modificated. Therefore, some researchers have focused on bit mapping-based self-embedding fragile watermarking algorithms. In bit mapping-based algorithms, the recovery bits obtained from whole image are scrambled and then embedded into the entire image according to embedding strategy.

With respect to the authentication process, self-embedding fragile watermarking algorithms can be categorized into two classes: block-wise schemes and pixel-wise schemes. In the block-wise authentication-based self-embedding fragile watermarking algorithms, the host image is divided into blocks and then the authentication bits generated for each block are embedded into the block itself. Meanwhile, in the pixel-wise authentication-based self-embedding fragile watermarking algorithms, the authentication information generated for each pixel values is embedded into the pixel itself.

Most of the self-embedding fragile watermarking algorithms use block-wise authentication approach. A block-wise authentication-based self-embedding fragile watermarking scheme which provides double chance for recovery of the detected manipulated areas was proposed by Lee and Lin [11]. In this method, two parity check bits generated for each 2×2 block were used for manipulation detection. Ansari et al. [1] proposed fragile watermarking method based on singular value decomposition (SVD) for manipulation detection and self-recovery. In this method, the trace of the singular matrix was used to generate the manipulation detection bits for each 4×4 block. An iterative restoration mechanism-based fragile watermarking method was presented by Bravo-Solorio et al. [2]. In this method, for each 8×8 block, five most significant bits (MSBs) of the block pixels and 160 reference bits were used to generate the authentication bits. Sing and Sing [22] proposed a self-embedding watermarking scheme based on discrete cosine transformation (DCT). Ten recovery bits and two authentication bits were generated from the five MSBs of the pixels for each 2×2 block. Qian et al. [16] presented a self-embedding fragile watermarking method with multi-level encoding, in which, according to smoothness degrees, different types of blocks were used for recovery bits generation. In this method, for each 8×8 block, 320 bits extracted from the five MSBs of the block and 160 recovery bits to be embedded into this block were fed into the hash function to generate 32 authentication bits. Qin et al. [17] presented a fragile watermarking method based on overlapping embedding strategy for manipulation detection and recovery. Block wise authentication and pixel-wise recovery schemes were collaborated in this method. According to complexity of the 3×3 blocks, 2, 3, or 4 authentication bits were embedded into the least significant bit (LSB) layer of the central pixel. Yang et al. [33] proposed halftone-based manipulation detection and content recovery method, in which watermark was embedded into the first LSBs of the reference image. 16 authentication bits for each 8×8 block were obtained from seven MSBs of the block and the 48 recovery bits. A self-embedding watermarking method based on discrete wavelet transform (DWT) and hierarchical manipulation detection was proposed by Tai and Liao [25]. In this method, DWT was used in recovery data generation process to reduce the smoothing blocking effect. For each 4×4 block, 32 watermark bits consisting of the 4 authentication bits and 28 recovery bits were used for manipulation detection and recovery. Sreenivas and Kamakshiprasa [24] presented image tamper detection methods, in which authentication bits were generated using chaotic maps. One of these methods used 12 authentication bits and the other used 4 authentication bits for each 2×2 block. Rhayma et al. [20] presented DWT and data representation thought combination-based semi-fragile self-embedding watermarking method. In this method, two copies of four authentication bits for each 32×32 block were embedded into approximation sub-band of the one-level DWT transformed image. Shehab et al. [21] proposed a fragile watermarking based on SVD for manipulation detection and recovery of the sensitive images in medical applications. Traces of singular matrices were used to generate 12 authentication bits for each 4×4 block, and then, average values of the five MSBs were used to generate recovery bits for each 2×2 block.

Block-wise authentication-based self-embedding fragile watermarking methods have a major problem. In authentication process of these methods, when one of the pixels in the block is detected to be manipulated, the entire block is marked as tampered. Therefore, pixel-wise authentication-based methods are more advantageous than block-wise authentication-based methods against some types of attacks such as salt and pepper noise adding. A pixel-wise authentication-based self-embedding fragile watermarking scheme was proposed by Lee et al. [10]. In this method, two authentication bits for each pixel were generated using five MSBs of the pixel. Since the pixel position information and the secret key were not used during the authentication bits generation phase, complex attacks such as collage, vector quantization, and swapping of pixels cannot be detected in this method. Also, Singh et al. [23] presented a dynamic domain-based pixel-wise authentication scheme with multi-level tamper detection. Four recovery bits generated from the five MSBs of the 1×2 block were embedded into two LSBs

of each DCT coefficient of the block in this method. Also, three authentication bits were generated for each pixel using content, location, and neighborhood. Then, authentication bits were embedded into the three LSBs of each pixel. However, using the three LSBs of each pixel for authentication bits embedding and two LSBs of the DCT coefficient of each block for recovery bits embedding decreased the quality of watermarked image.

In this paper, to overcome the above defect found in block-wise and pixel-wise authentication-based self-embedding fragile watermarking methods, we propose a new pixel-wise authentication-based self-embedding fragile watermarking method. In this proposed method, reference image is divided into four equal parts denoted as a main blocks. Then, each part is subdivided into 2×4 or 4×2 sub-blocks according to block type. Average values of the sub-blocks are calculated and combined as recovery bits of each main block. Then, recovery bits to be embedded into each main block are constructed by merging the half of the recovery bits of the two main blocks in the other half of the image. After the recovery bits generation process, the five MSBs of the pixel, recovery bit, and two pixel position bits are used to generate two authentication bits for each pixel. Then, authentication bits are embedded into the first and second LSBs of the pixel, while recovery bit is embedded into the third LSB of the pixel. To demonstrate the performance of the proposed method, pixel-based attacks have been applied to the watermarked image. Also, different sized cropping attacks have been used to evaluate the success of the method.

The rest of paper is organized as follows. Section 2 contains the proposed method. Section 3 shows the experimental results. The paper is concluded in Section 4.

2 Proposed method

We have designed a pixel-wise authentication-based self-embedding fragile watermarking method that not only verifies the integrity of the image, but also detects and recovers the manipulated areas. For this purpose, authentication watermark is used to detect and localize the manipulated areas, while the recovery watermark is used to recover tampered areas. The proposed scheme consists of five main processes: determination of the block type and generation of the pixel position bits, recovery bits generation process, authentication bits generation and watermark embedding process, manipulation detection process, and recovery process.

2.1 Determination of the block type and generation of the pixel position bits

In the proposed recovery bits embedding process, horizontal (2×4) or vertical (4×2) blocks are used for recovery bits generation. Suitable block type for each original image is determined separately to obtain high visual quality of recovered image. The determination of block type is shown in Fig. 1. First, the original image is divided into blocks of 2×4 sizes and the average value of each block is calculated. Then, the average value is enlarged to 2×4 block size. Finally, the fully horizontally recovered image is obtained using these enlarged blocks. The same operations are also performed using the 4×2 block size to obtain fully vertically recovered image. After the generation of fully recovered images, Peak Signal-to-Noise Ratio (PSNR) values between the original image and the fully recovered images are calculated using Eqs. (1) and

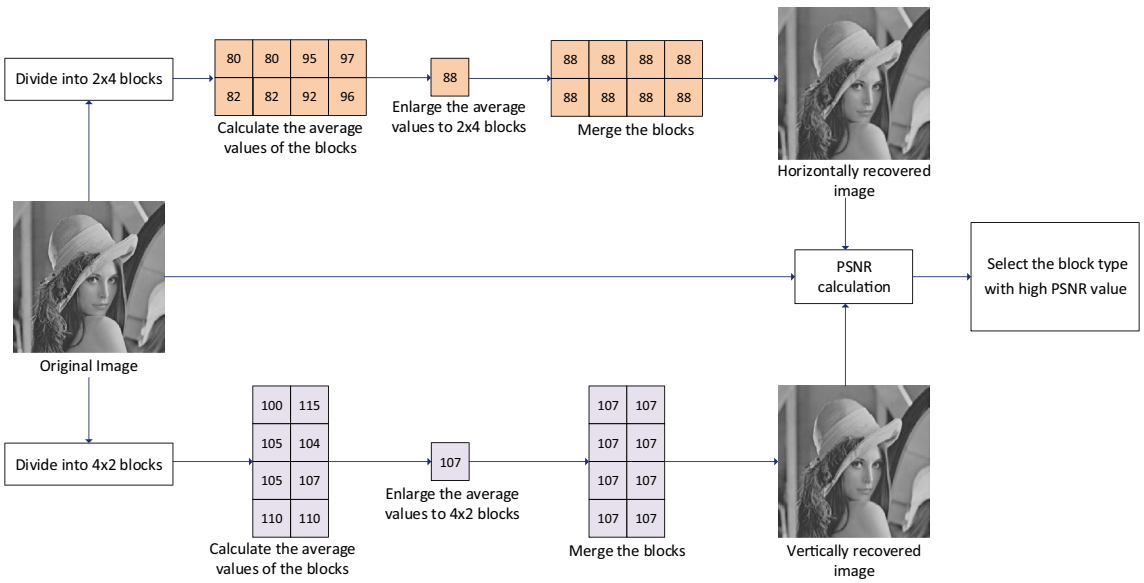

Fig. 1 Determination of block type for original image

(2). The PSNR values, which are used to quantify eminence of fully recovered images, are calculated as follows [19]:

$$PSNR(I^r, I^t) = 10 \times log_{10}\left(255^2 / MSE(I^r, I^t)\right) \quad (1)$$

$$MSE = 1/S \times T \sum_{i=1}^{S} \sum_{j=1}^{T} (I^r_{(i,j)} - I^t_{(i,j)})^2, \quad (2)$$

where I^r and I^t are the reference and the test images size of $S \times T$, and $I^r_{(i,j)}$ and $I^t_{(i,j)}$ denote the pixel values at position (i, j) of the reference and test images [20].

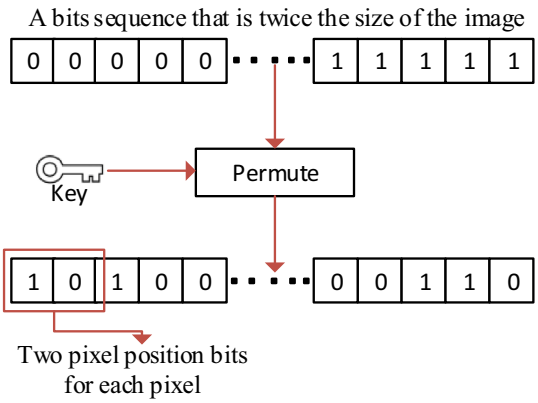

Fig. 2 Generation of the pixel position bits

In the proposed authentication bits generation and embedding process, to detect pixel position swapping attacks, two pixel position bits are used for each pixel. The generation of the pixel position bits is shown in Fig. 2. A sequence of bits is created in which the number of zeros and ones is equal, and the size is twice the number of pixels in the image. Then, this pixel position sequence is permutated with a secret key. To be effective against complex attacks, different keys must be used for different images.

2.2 Recovery bits generation process

In this process of the proposed method, a compressed version of the original image is generated for watermark embedding process. The block diagram of the recovery bits generation process is shown in Fig. 3. The following steps demonstrate a detailed description of the recovery bits generation process:

1. The original image, I^o, with size of $S \times T$ is divided into four non-overlapping main blocks, $MB_{(i)}(i = 1, 2, \ldots, 4)$, with size of $K \times L$.
2. Each main block, $MB_{(i)}$, is subdivided into 4×2 or 2×4 sub-blocks, $MB_{(i,j)}(j = 1, 2, \ldots, K \times L/8)$, according to determined block type.
3. The 8-bit average values, $M_{(i,j)}$, of the sub-blocks are calculated and combined as recovery data of each main block, $M_{(i)}$.

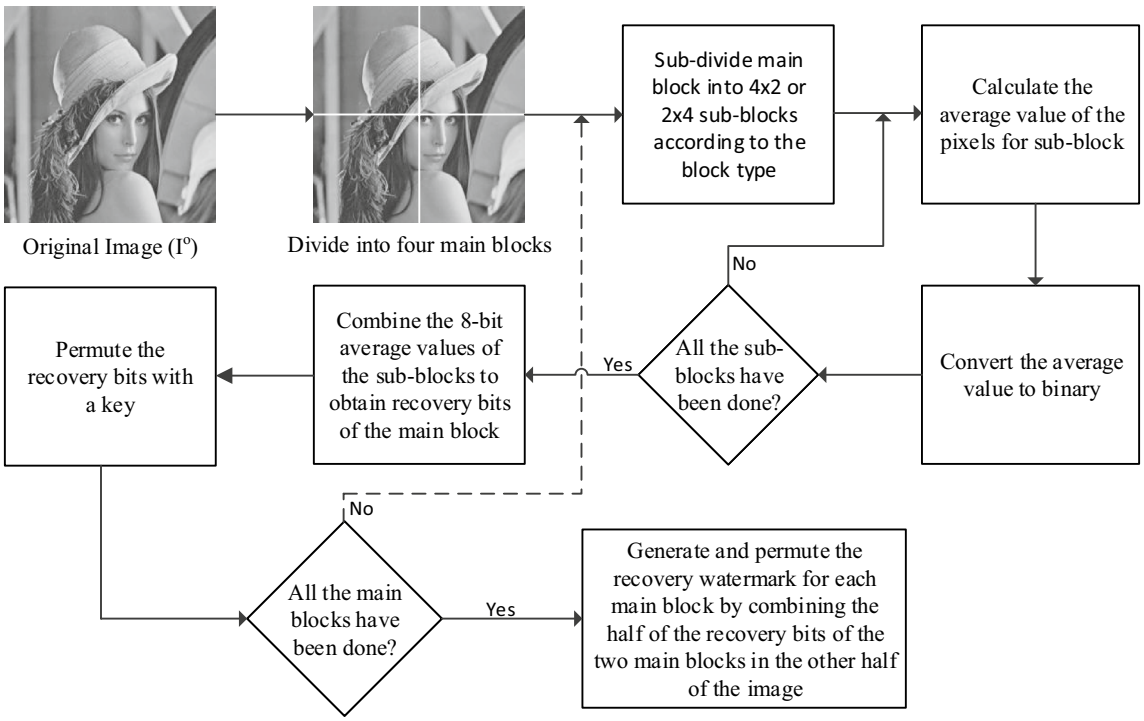

Fig. 3 Block diagram of the proposed recovery bits generation process

4. The generated recovery bits for each main block, $M_{(i)}$, are permutated with a key.
5. Recovery bits to be embedded into each main block, $CR_{(i)}$, are generated by combining the half of the recovery bits of the two main blocks in the other half of the image. In other words, the recovery bits of each main block are spread to two main blocks in the other half of the image.
6. Combined recovery bits to be embedded into each main block, $CR_{(i)}$, are permutated with a key.
7. Recovery watermarks, $RW_{(i)}$, to be embedded in each main block are obtained for watermark embedding process.

2.3 Authentication bits generation and watermark embedding process

In this process of the proposed method, for each pixel of the image, two authentication bits are generated and then embedded with the one recovery bit using LSB substitution. In authentication bits generation phase, XOR operation is used to generate one of the authentication bits. For each pixel, five MSBs of the pixel, one recovery bit and two pixel position bits are used for generating first authentication bit, as shown in Fig. 4. XOR is the most basic and most widely used hashing operation. Also, using hash function such as MD5 or SHA to generate 2 bit hash value is both unnecessary and time-consuming. On the other hand, second authentication bit for each pixel is generated by inversing the first generated authentication bit. Inverse operation is used to detect the attacks such as salt and pepper adding, and cropping in this proposed method. The block diagram of the authentication bits generation and watermark embedding process is shown in Fig. 5. The following steps demonstrate a detailed description of this process:

1. The original image, I^o, with size of $S \times T$ is divided into four non-overlapping main blocks, $MB_{(i)}(i = 1, 2, \ldots, 4)$, with size of $K \times L$.
2. Main block, $MB_{(i)}$, is subdivided into 4×2 or 2×4 sub-blocks, $MB_{(i,j)}(j = 1, 2, \ldots, K \times L/8)$, according to determined block type.
3. Each pixel value, $P_{(e)}(e = 1, 2, \ldots, 8)$, of the sub-block, $MB_{(i,j)}$, is converted to binary.
4. Recovery bit for the sub-block, $RW_{(i,j)}$, is embedded into the third LSBs of the pixels.
5. Authentication bits for each pixel, $AU1_{(e)}$ and $AU2_{(e)}$, are generated using six MSBs of the pixel and pixel position bits, $PPB^k_{(e)}(k = 1, 2)$, as shown in Fig. 4, as follows:

   i. $A1$ is generated from XOR result of first three MSBs of the pixel and first pixel position bit, PPB^1, using Eq. 3:

   $$A1 = \sim (((MSB1) \oplus (MSB2)) \oplus ((MSB3) \oplus (PPB^1))). \quad (3)$$

   ii. $A2$ is generated from XOR result of the fourth, fifth, and sixth (recovery bit) MSBs of the pixel, and second pixel position bit, PPB^2, using Eq. 4:

   $$A2 = \sim (((MSB4) \oplus (MSB5)) \oplus ((MSB6) \oplus (PPB^2))). \quad (4)$$

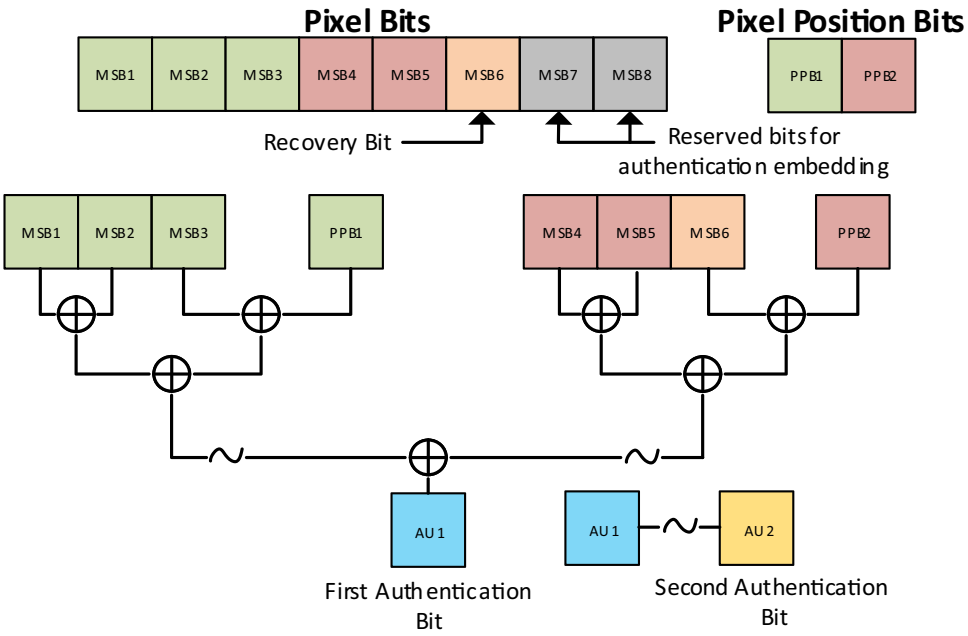

Fig. 4 Generation of the two authentication bits for each pixel

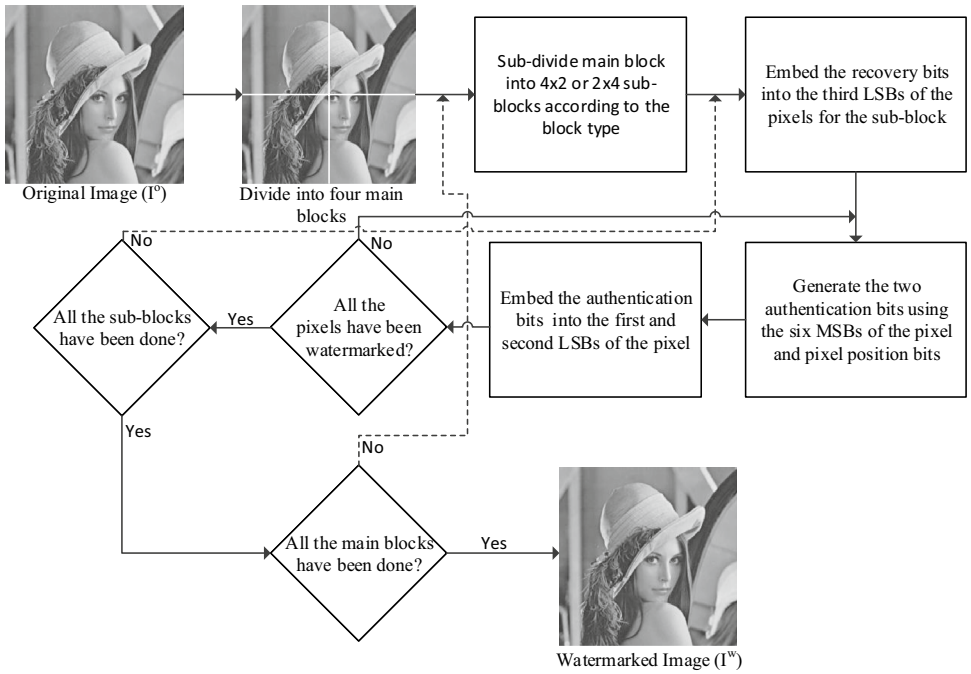

Fig. 5 Block diagram of the proposed authentication bits generation and watermark embedding process

iii. First authentication bit, $AU1$, is generated by XOR-ing $A1$ and $A2$ as follows:

$$AU1 = A1 \oplus A2. \tag{5}$$

iv. Second authentication bit, $AU2$, is generated from the inverse result of $AU1$ using Eq. 6.

$$AU2 = \sim (AU1). \tag{6}$$

6. Generated authentication bits for each pixel, $AU1_{(e)}$, $AU2_{(e)}$, are embedded into the first and second LSBs of the pixels.
7. 3–6 steps are repeated for each sub-block.
8. 2–7 steps are repeated for each main block.
9. Watermarked image, I^w, is obtained.

2.4 Manipulation detection process

In manipulation detection process, to obtain manipulation map, three-level authentication is applied to the image. In first authentication level, extracted authentication bits are compared with re-generated authentication bits in the sub-block. Then, manipulated pixels are marked as tampered. In second authentication level, for each sub-block which has tampered pixel, the overlapping 2×2 blocks are checked. If 3 or more pixels are determined as manipulated pixels in the 2×2 overlapping blocks, all pixels in this overlapping blocks are marked as tampered. An example of the second authentication level is shown in Fig. 6. In third authentication level, all pixels in the sub-block are marked as tampered if 5 or more pixels are determined as manipulated pixels in this sub-block. An example of the third authentication level is shown in Fig. 7. The block diagram of the manipulation detection process is shown in Fig. 8. The following steps demonstrate a detailed description of manipulation detection process:

1. The manipulated image, I^m, with size of $S \times T$ is divided into four non-overlapping main blocks, $MB_{(i)}(i = 1, 2, \ldots, 4)$, with size of $K \times L$.

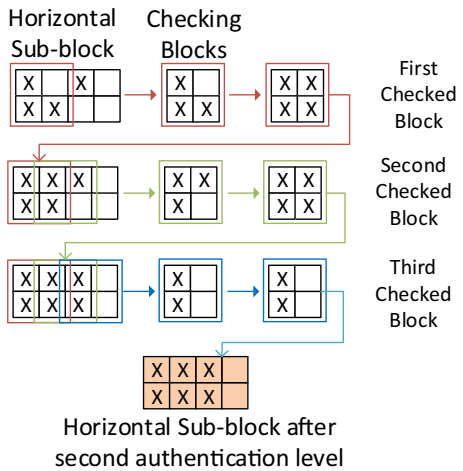

Fig. 6 An example of the second authentication level

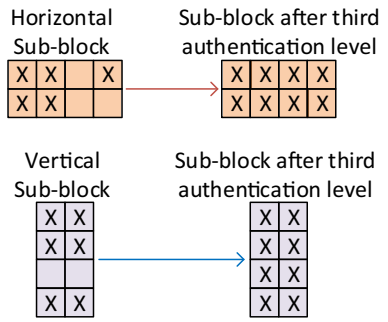

Fig. 7 An example of the third authentication level

2. Main block, $MB_{(i)}$, is subdivided into 4×2 or 2×4 sub-blocks, $MB_{(i,j)}(j = 1, 2, \ldots, K \times L/8)$, according to determined block type.
3. Each pixel value, $P_{(e)}(e = 1, 2, \ldots, 8)$, of the sub-block, $MB_{(i,j)}$, is converted to binary, $BP_{(e)}$.
4. Authentication bits for each pixel, $RAU1_{(e)}$, $RAU2_{(e)}$, are re-generated using Eqs. (3–6) like authentication bits generation process.
5. Comparison bits, $EAU1_{(e)}$, $EAU2_{(e)}$, are extracted from first and second LSBs of the pixel using LSB substitution.
6. Extracted authentication bits, $EAU1_{(e)}$, $EAU2_{(e)}$, are compared with re-generated authentication bits, $RAU1_{(e)}$, $RAU2_{(e)}$.
7. Manipulated pixels are marked as a tampered according to the comparison result.
8. Sub-block, $MB_{(i,j)}$, is divided into overlapping 2×2 check blocks, $C_{(f)}(f = 1, 2, 3)$.
9. All pixels in the overlapping blocks are marked as tampered if there are 3 or more manipulated pixels in this 2×2 overlapping blocks.
10. All pixels in sub-block are marked as tampered if there are 5 or more manipulated pixels in this sub-block.
11. 3–10 steps are repeated for each sub-block.
12. 2–11 steps are repeated for each main block.
13. Manipulation detected image, I^{md}, and manipulation map, $Mmap$, are obtained.

2.5 Recovery process

In this process, recovery bits are extracted from third LSBs of the pixels, and then, fully recovery image is constructed using extracted bits. According to manipulation map,

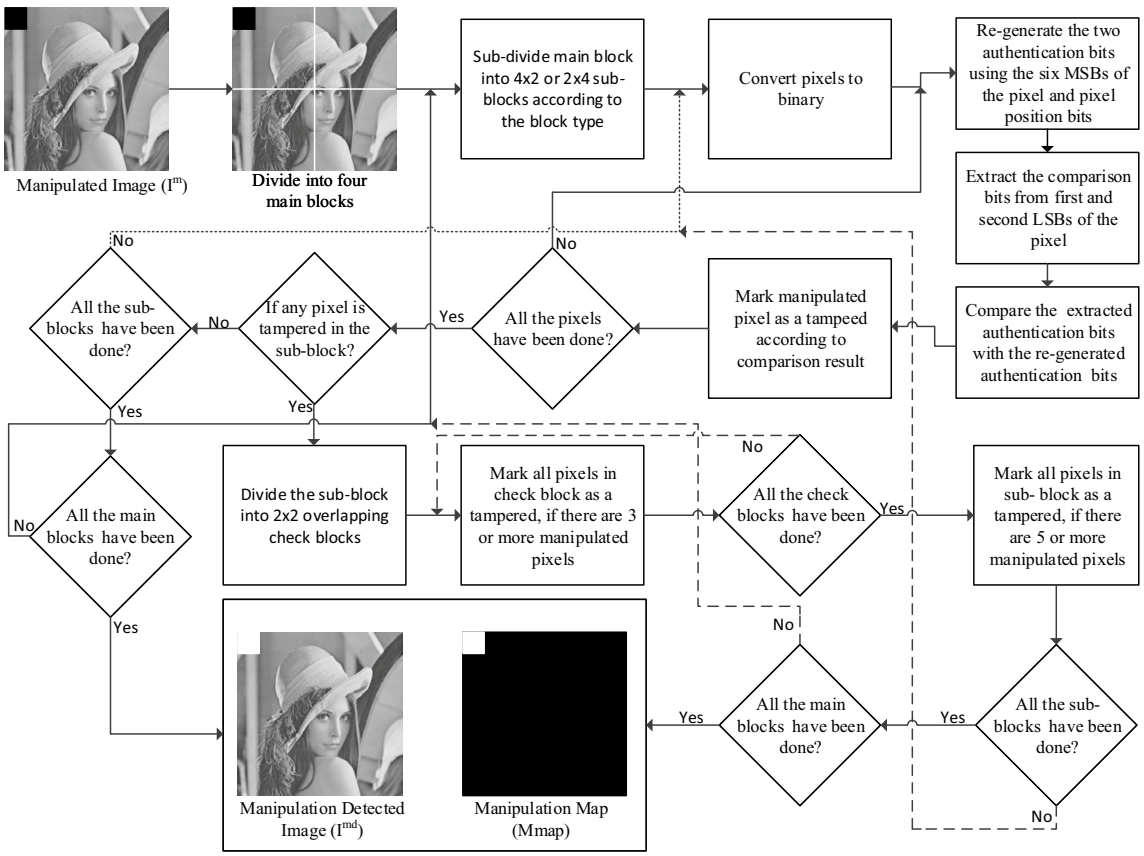

Fig. 8 Block diagram of the proposed manipulation detection process

manipulated pixels are replaced with recovered pixels. The block diagram of the recovery process is shown in Fig. 9. The following steps demonstrate a detailed description of manipulation detection process:

1. The manipulated image, I^m, with size of $S \times T$ is divided into four non-overlapping main blocks, $MB_{(i)}(i = 1, 2, \ldots, 4)$, with size of $K \times L$.
2. Each main blocks, $MB_{(i)}$, is subdivided into 4×2 or 2×4 sub-blocks, $MB_{(i,j)}(j = 1, 2, \ldots, K \times L/8)$, according to determined block type.
3. Each pixel value, $P_{(e)}(e = 1, 2, \ldots, 8)$, of the sub-block, $MB_{(i,j)}$, is converted to binary.
4. Recovery bits, $R_{(i,j)}$, are extracted from third LSBs of the pixels, $P_{(e)}$, for each sub-block.
5. The recovery bits extracted from each sub-block, $R_{(i,j)}$, are combined for each main block, $R_{(i)}$.
6. The recovery bits extracted from each main block, $R_{(i)}$, are permutated with a key.
7. Recovery data of each main block, $CR_{(i)}$, are generated with combining half of the recovery bits extracted from the two main blocks in the other half of the manipulated image.
8. Combined recovery data of each main block, $CR_{(i)}$, are permutated with a key.
9. Recovery watermark bits, $RW_{(i)}$, of each main block are divided to rebuild the sub-blocks.

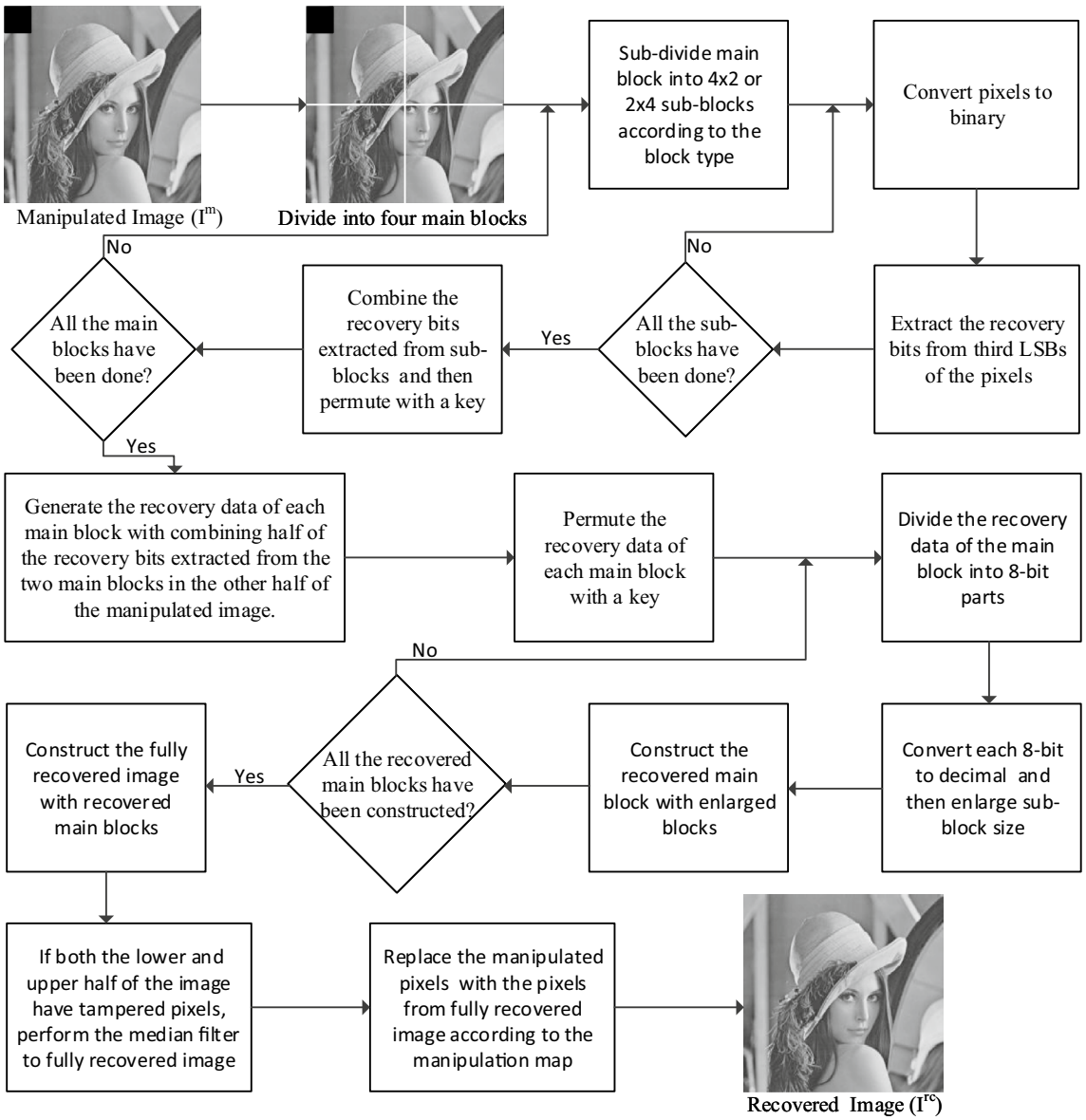

Fig. 9 Block diagram of the proposed recovery process

10. The divided recovery bits are converted to decimal and then enlarged to sub-block size.
11. Enlarged blocks are constructed to obtain fully recovered image.
12. If both the lower and upper half of the image have tampered pixels, median filter with 5×5 neighborhood size is performed to fully recovered image.
13. Manipulated pixels are replaced with the pixels from fully recovered image according to manipulation map.
14. Recovered image, I^{rc}, is obtained.

3 Experimental results

A set of standard images such as Lena are used to the test the performance of the proposed method. The quality assessment between the reference image and watermarked image is measured by PSNR and Structural similarity index (SSIM) metrics. The PSNR and SSIM metrics are the indicators of perceptual quality of the images [18, 26]. PSNR values between the reference and the test images are calculated by Eqs. (1) and (2). SSIM values between the reference and the test images are defined by Eq. (7) as follows:

$$SSIM = \left(\frac{(2\mu_{(I^r)}\mu_{(I^t)} + C_1)(2\sigma_{(I^r,I^t)} + C_2)}{(\mu^2_{(I^r)} + \mu^2_{(I^t)} + C_1)((\sigma^2_{(I^r)} + \sigma^2_{(I^t)} + C_2)} \right), \tag{7}$$

where $\mu_{(I^r)}$ and $\mu_{(I^t)}$ are the local means of the reference and test image, $\sigma_{(I^r,I^t)}$ is the co-variation of the reference and test images, and $\sigma_{(I^r)}$ and $\sigma_{(I^t)}$ are the standard deviations of the reference and test images. C_1 and C_2 are the constants [28].

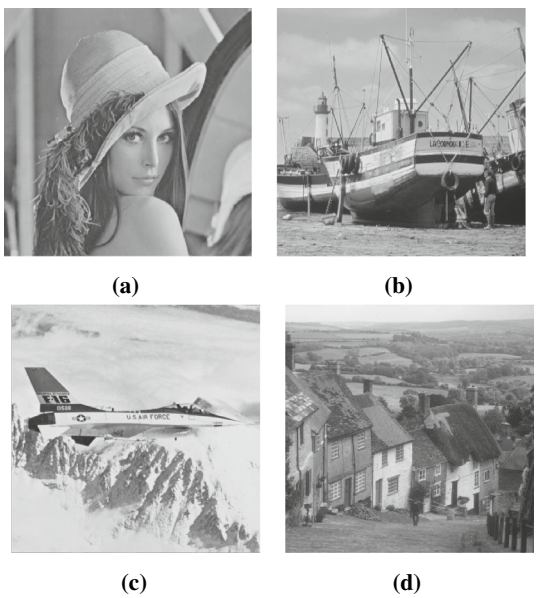

Fig. 10 Reference images : (**a**) Lena, (**b**) SailBoat, (**c**) Airplane, and (**d**) Goodhill

Lena, Sailboat, Airplane, and Goodhill images size of 512×512 are used as reference images to test the performance of the proposed method. The reference and watermarked images are illustrated in Figs. 10 and 11, respectively. The PSNR and SSIM results between reference and the watermarked images are shown in Table 1. As seen from the table, the PSNR and SSIM values of the watermarked images are higher than 38 *dB* and 0.92, respectively. Due to length of the watermark using for each pixel, the PSNR values of the watermarked images are found to be about 38 *dB*. It is clear from the results that the visual qualities of the reference images have been protected during watermarking process. On the other hand, block types of Lena, Sailboat and Goodhill images are determined as "Vertical", while block type of Airplane image is determined as "Horizontal" for recovery bits generation process.

The detection achievements of the proposed method and block-wise authentication-based methods are measured by Manipulation pixels (MP), Detection pixels (DP), False detection pixels (FDP), and False positive rate (FPR) metrics. FPR is the ratio of non-manipulated pixels detected

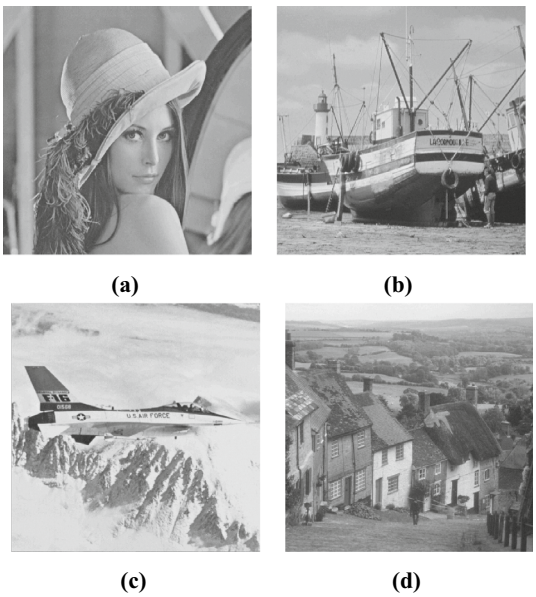

Fig. 11 Watermarked images : (**a**) Lena, (**b**) SailBoat, (**c**) Airplane, and (**d**) Goodhill

Table 1 The PSNR and SSIM results of the watermarked images

Image	Lena	SailBoat	Airplane	Goodhill
PSNR (*dB*)	38.3545	38.3829	38.3545	38.3418
SSIM	0.9260	0.9380	0.9382	0.9521
Block type	Vertical	Vertical	Horizontal	Vertical

Table 2 The MP, DP, FDP, and FPR results of the methods against salt and pepper noise attacks

Methods	Metrics	Noise density			
		0,01	0,02	0,05	0,1
Proposed	MP	2663	5204	13122	26258
	DP	2663	5204	13122	26258
	FDP	0	0	0	0
	FPR	0%	0%	0%	0%
2x2 block based	MP	2663	5204	13122	26258
	DP	10432	20220	48816	90132
	FDP	7769	15016	35694	63874
	FPR	2,9636%	7,7281%	13,616%	24,3659%
4x4 block based	MP	2663	5204	13122	26258
	DP	39312	72368	147232	212800
	FDP	36649	67164	134110	186542
	FPR	13,9804%	25,6210%	51,1589%	71,1601%
8x8 block based	MP	2663	5204	13122	26258
	DP	125632	191424	251520	261696
	FDP	122969	186220	238398	235438
	FPR	46,9089%	71,0372%	90,9416	89,8124

as manipulated pixels [20]. FPR values of the methods are defined by Eq. 8 as follows:

$$FPR = \frac{number\ of\ authentic\ pixels\ detected\ as\ manipulated}{number\ of\ the\ pixels\ in\ manipulated\ image} \times 100. \tag{8}$$

The performance of the 2×2, 4×4, and 8×8 block-wise authentication-based methods and our proposed method against pixel-based attacks on the Lena image has been evaluated using salt and pepper noise adding attacks with noise density of "0.01", "0.02", "0.05", and "0.1". The MP, DP, FDP, and FPR results of the methods are shown in Table 2 for detection of the salt and pepper noise adding attacks. As can be seen from the table, as the noise density and block size increase, the FPR results of the block-wise authentication-based methods increase. However, the proposed method has completely detected all the manipulated pixels without affected by noise density ratio. Comparison between the detection processes of the block-wise authentication-based methods and our proposed method against salt and pepper noise adding attack with noise density of "0.05" which affects approximately 5% of the pixels has been demonstrated in Fig. 12. In block-wise authentication-based methods, even a pixel in the block is detected as manipulated, and the entire block is marked as attacked. Therefore, it is clear from the Fig. 12 that, due to pixel-wise authentication process, the proposed method is more successful than the block-wise authentication-based methods.

Also, different numbers of line type cropping attacks size of 512×11 have been applied to Goodhill image for comparison of the methods. The MP, DP, FDP, and FPR results of the methods against line type cropping attacks are given in Table 3. Comparison between the detection processes of the 2×2, 4×4 and 8×8 block-wise authentication-based methods and our proposed method against two line type cropping attacks size of 512×11 is demonstrated in Fig. 13. It is evident from Table 3 that as the

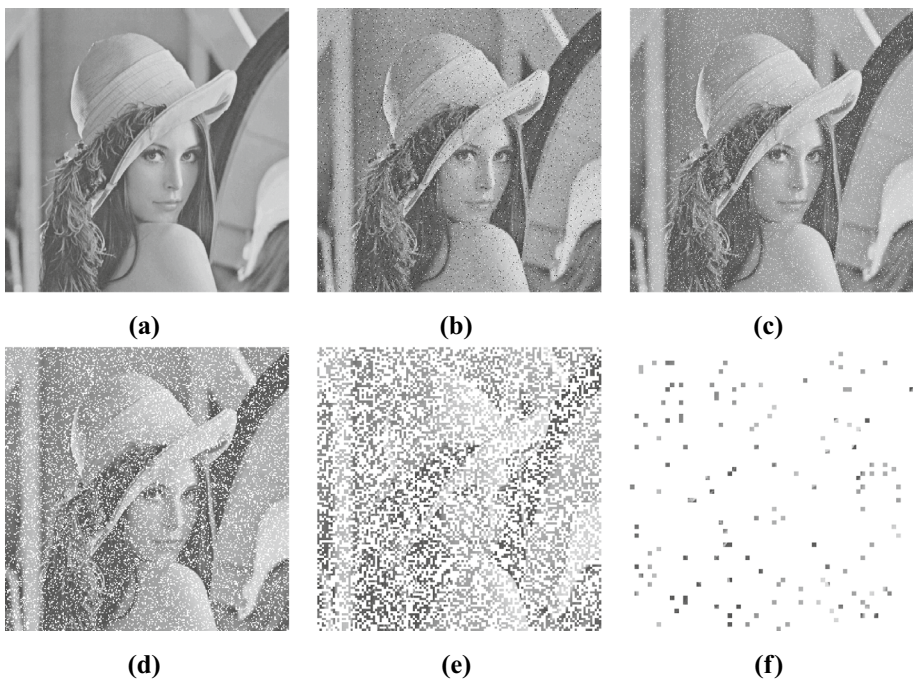

Fig. 12 Detection process comparison for salt and pepper noise adding attack with noise density of "0,05": (**a**) Watermarked Lena, (**b**) Salt and pepper noise added Lena, (**c**) Proposed method, (**d**) 2×2 block-wise method, (**e**) 4×4 block-wise method, and (**f**) 8×8 block-wise method

Table 3 The MP, DP, FDP, and FPR results of the methods against line cropping attacks

Methods	Metrics	Number of lines			
		1	2	3	4
Proposed	MP	5632	11264	16896	22528
	DP	5632	11264	16896	22528
	FDP	0	0	0	0
	FPR	0%	0%	0%	0%
2x2 block based	MP	5632	11264	16896	22528
	DP	6144	12288	18432	24576
	FDP	512	1024	1536	2048
	FPR	0,1953%	0,3906%	0,5859%	0,7812%
4x4 block based	MP	5632	11264	16896	22528
	DP	8192	16384	24576	32768
	FDP	2560	5120	7680	10240
	FPR	0,9765%	1,9531%	2,9296%	3,9062%
8x8 block based	MP	5632	11264	16896	22528
	DP	12288	24576	36864	49152
	FDP	6656	13312	19968	26624
	FPR	2,5390%	5,0781%	7,6171%	10,1562%

numbers of line cropping attack increase, the FPR results of the block-wise authentication-based methods increase. However, for all the different number of line cropping attacks, the proposed method has completely detected all the manipulated pixels. FPR results of the proposed method are obtained as zero against all line cropping attacks. Also, as can be seen from Fig. 13 that our pixel-wise authentication-based method is more efficient than the block-wise authentication-based methods against two line cropping attacks.

To demonstrate the performance comparison of our method with the other pixel-wise methods [10, 23], cropping and exchange of the content attacks are applied to Lena and Barbara images, as shown in Figs. 14 and 15, respectively. It can be seen from Fig. 14 that the proposed method and Lee et al.'s [10] method have detected entire of the manipulated area. Although Singh et al.'s method [23] detects the manipulated area, it cannot detect all tampered pixels. As a result, the proposed method and Lee et al.'s [10] method are more successful than Singh et al.'s [23] method against cropping attack. On the other hand, for exchange of content attack, it is clear from Fig. 15 that Singh et al.'s [23] method and the proposed method have detected the manipulated areas. However, Lee et al.'s [10] method has not detected the manipulated areas. Pixel position information and secret key are not used in Lee et al.'s [10] method, so that the complex attacks such as exchange of content attack cannot be detected. Also, although our method detects the manipulated areas, it cannot detect all the tampered pixels. Therefore, Singh et al.'s [23] method is more efficient than the proposed method and Lee et al.'s [10] method against exchange of content attack. Also, proposed method is more effective than Lee et al.'s [10] method against exchange of content attack.

Recovery ability of the proposed method against pixel-based attack has been evaluated using salt and pepper noise adding attack with noise density of "0.1". Watermarked, manipulated, manipulation detected, and recovered Lena images are shown in Fig. 16, respectively. It can be seen

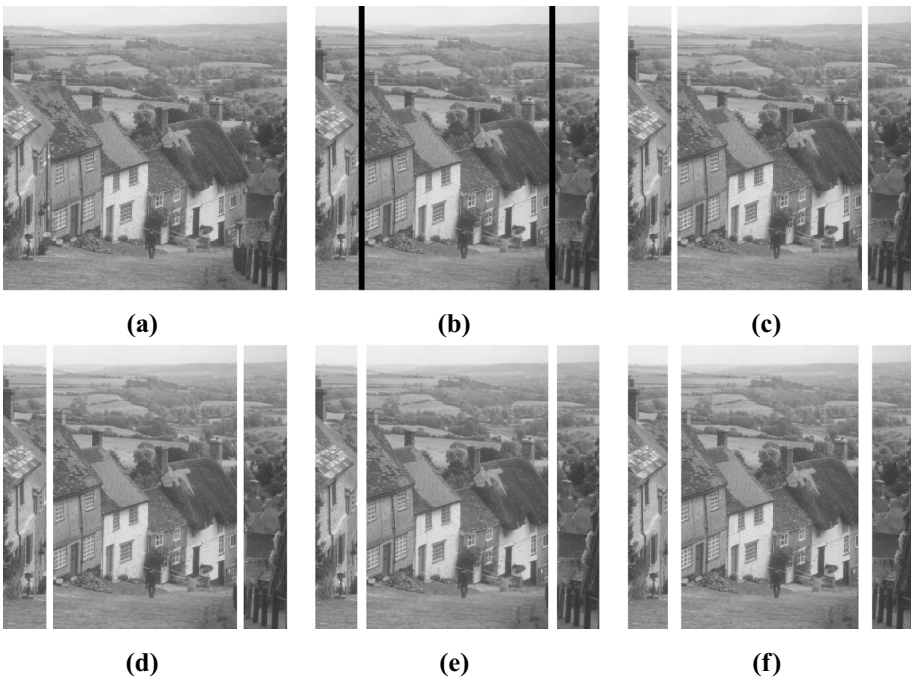

Fig. 13 Detection process comparison for two line type cropping attack: (**a**) Watermarked Lena, (**b**) Line type cropping attacked Lena, (**c**) Proposed method, d) 2×2 block-wise method, (**e**) 4×4 block-wise method, and (**f**) 8×8 block-wise method

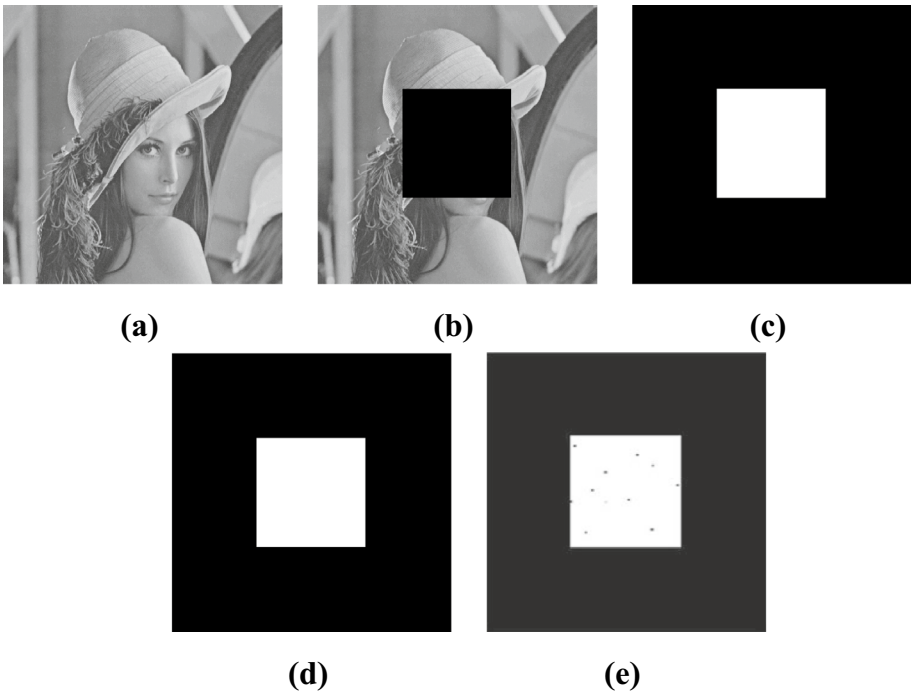

Fig. 14 Detection process comparison of pixel-wise methods against 15% cropping attack: (**a**) Watermarked Lena, (**b**) Manipulated Lena, (**c**) Proposed method, (**d**) Lee et al.'s [10] method, and (**e**) Singh et al.'s [23] method

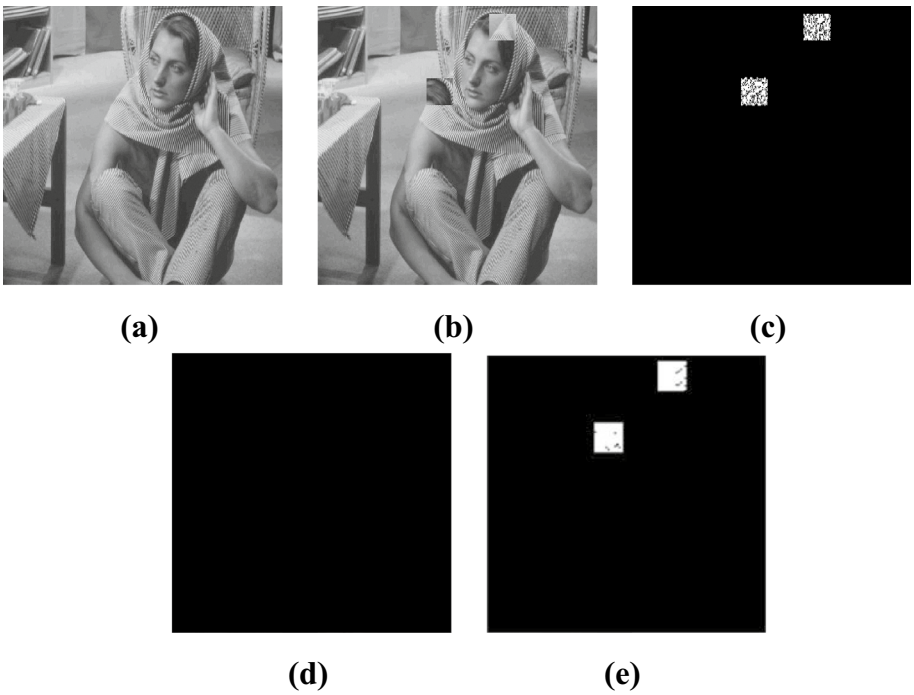

Fig. 15 Detection process comparison of pixel-wise methods against exchange of content attack: (**a**) Watermarked Barbara, (**b**) Manipulated Barbara, (**c**) Proposed method, (**d**) Lee et al.'s [10] method, and (**e**) Singh et al.'s [23] method

from the figure that salt and pepper noises have been successfully localized and recovered. The PSNR result of the recovered Lena image is obtained as 36.1632 *dB* and the SSIM result of the recovered Lena image is get as 0.9594.

To demonstrate the manipulation detection and recovery of the proposed method, 25% and 50% cropping attacks are applied to the watermarked Airplane and Sailboat images, as shown in Figs. 17 and 18. Manipulation detected and recovered images for corresponding attacked images are also shown in Figs. 17 and 18. As can be seen from the figures that both 25% and 50% cropping attacks have been successfully localized and recovered by the proposed method. The PSNR results of the Airplane and Sailboat images are obtained as 32.7422 *dB* and 31.5430 *dB*, respectively.

Fig. 16 Results of salt and pepper attack with noise density of "0.1": (**a**) Watermarked, (**b**) Manipulated, (**c**) Manipulation localized, and (**d**) Recovered Lena

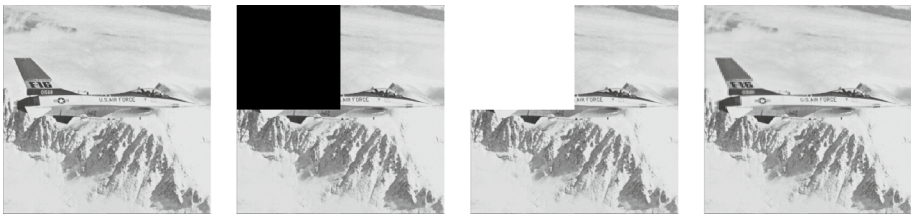

Fig. 17 Results of 25% cropping attack: (**a**) Watermarked, (**b**) Manipulated, (**c**) Manipulation localized, and (**d**) Recovered Airplane

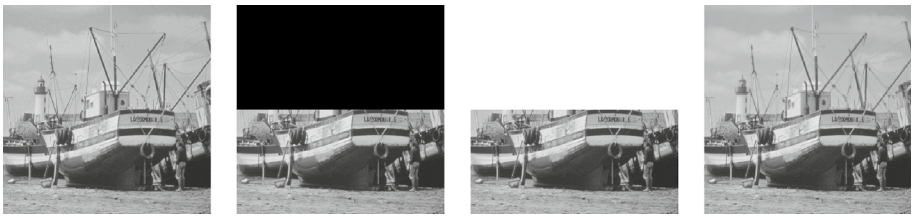

Fig. 18 Results of 50% cropping attack: (**a**) Watermarked, (**b**) Manipulated, (**c**) Manipulation localized, and (**d**) Recovered Goodhill

Table 4 The PSNR and SSIM results for different manipulation rated cropping attacks

Image	Performance of recovered image	Manipulation rate				
		10%	20%	30%	40%	50%
Lena	PSNR	45.2321	40.0604	37.5577	34.9506	33.8520
	SSIM	0.9821	0.9609	0.9399	0.9166	0.8969
SailBoat	PSNR	44.9235	38.9909	36.8332	33.3924	31.5430
	SSIM	0.9822	0.9643	0.9457	0.9204	0.8951
Airplane	PSNR	41.4692	39.1642	37.1838	32.8989	30.9820
	SSIM	0.9803	0.9621	0.9447	0.9245	0.9055
Goodhill	PSNR	42.5105	37.3784	34.9900	32.3899	31.3536
	SSIM	0.9791	0.9485	0.9173	0.8797	0.8462

On the other hand, to evaluate the proposed method, different manipulation rated cropping attacks have been applied to the watermarked Lena, Sailboat, Airplane, and Goodhill images. Table 4 shows the PSNR and SSIM results of the recovered images. The best PSNR results have been obtained from the Lena image. For all the 50% cropped images, PSNR and SSIM values of the recovered images are higher than 30 *dB* and 0.84, respectively.

4 Conclusions

According to the authentication process, self-embedding fragile watermarking methods can be classified as block-wise schemes and pixel-wise schemes. In block wise authentication-based methods, the whole image block is marked as tampered if one of the pixels in the block

is determined as manipulated. Therefore, accuracy rate of the block wise authentication-based methods is lower than the pixel-wise authentication-based methods against pixel-based attacks. In this work, we have developed a novel pixel-wise authentication-based self-embedding fragile watermarking method to reduce the false detection rate of pixel-based attacks. In this method, to obtain high visual quality of recovered image, vertical (4×2) or horizontal (2×4) block type for the image is determined for watermarking process. Also, two pixel position bits for each pixel of the image are generated to detect the attacks such as pixel position swapping. After these processes, original image is divided into four main blocks. Recovery bits to be embedded into each main block are generated from the two main blocks in the other half of the image. In other words, the recovery bits of each main block are scrambled and embedded into the main blocks of the other half of the image. Then, for each pixel, two authentication bits are generated using five MSBs of the pixel, recovery bit to be embedded into the pixel and two pixel position bits. Recovery bits are embedded into the third LSBs of the pixels, while authentication bits are embedded into the first and second LSBs of the pixels.

The accomplishment of the proposed method has been demonstrated by applying pixel-based attacks and different sized cropping attacks to the watermarked images. The manipulation detection of the proposed method under pixel-based attacks is found to be fairly superior to some block-wise authentication-based methods. Besides, in case of different manipulation rated cropping attacks, the proposed method builds a good estimation of the original image. The quality of the watermarked images is sufficient with an about 38 *db*.

References

1. Ansari, I.A., Pant, M., Ahn, C.W.: Svd based fragile watermarking scheme for tamper localization and self-recovery. Int. J. Mach. Learn. Cybernet. **7**(6), 1225–1239 (2016)
2. Bravo-Solorio, S., Calderon, F., Li, C.T., Nandi, A.K.: Fast fragile watermark embedding and iterative mechanism with high self-restoration performance. Digital Sig. Process. **73**, 83–92 (2018)
3. Cao, F., An, B., Wang, J., Ye, D., Wang, H.: Hierarchical recovery for tampered images based on watermark self-embedding. Displays **46**, 52–60 (2017)
4. Chen, W.C., Wang, M.S.: A fuzzy c-means clustering-based fragile watermarking scheme for image authentication. Expert Systems with Applications 36(2, Part 1):1300 – 1307 (2009)
5. Chuang, J.C., Hu, Y.C.: An adaptive image authentication scheme for vector quantization compressed image. J. Vis. Commun. Image Represent. **22**(5), 440–449 (2011)
6. Chun-Shien, Lu, Liao, H.M.: Structural digital signature for image authentication: an incidental distortion resistant scheme. IEEE Trans. Multimedia **5**(2), 161–173 (2003)
7. Gul, E., Ozturk, S.: A novel hash function based fragile watermarking method for image integrity. Multimedia Tools Appl. **78**(13), 17701–17718 (2019)
8. Huang, R., Liu, H., Liao, X., Sun, S.: A divide-and-conquer fragile self-embedding watermarking with adaptive payload. Multimedia Tools Appl. **78**(18), 26701–26727 (2019)
9. Kim, C., Shin, D., Yang, C.N.: Self-embedding fragile watermarking scheme to restoration of a tampered image using ambtc. Personal Ubiquitous Comput. **22**(1), 11–22 (2018)
10. Lee, C.F., Shen, J.J., Chen, Z.R., Agrawal, S.: Self-embedding authentication watermarking with effective tampered location detection and high-quality image recovery. Sensors **19**(10), 2267 (2019)
11. Lee, T.Y., Lin, S.D.: Dual watermark for image tamper detection and recovery. Pattern Recognit. **41**(11), 3497–3506 (2008)
12. Lin, P., Lee, J., Chang, C.: Dual digital watermarking for internet media based on hybrid strategies. IEEE Trans. Circuits Syst. Video Technol. **19**(8), 1169–1177 (2009)
13. Lin, P.L., Hsieh, C.K., Huang, P.W.: A hierarchical digital watermarking method for image tamper detection and recovery. Pattern Recognit. **38**(12), 2519–2529 (2005)
14. Martino, F.D., Sessa, S.: Fragile watermarking tamper detection with images compressed by fuzzy transform. Inform. Sci. **195**, 62–90 (2012)
15. Puhan, N.B., Ho, A.T.: Secure authentication watermarking for localization against the holliman-memon attack. Multimedia Syst. **12**(6), 521–532 (2007)
16. Qian, Z., Feng, G., Zhang, X., Wang, S.: Image self-embedding with high-quality restoration capability. Digital Signal Process. **21**(2), 278–286 (2011)
17. Qin, C., Ji, P., Zhang, X., Dong, J., Wang, J.: Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. Signal Process. **138**, 280–293 (2017)
18. Rajkumar, R., Vasuki, A.: Reversible and robust image watermarking based on histogram shifting. Cluster Comput. **22**(5), 12313–12323 (2019)
19. Rayachoti, E., Tirumalasetty, S., Prathipati, S.C.: Slt based watermarking system for secure telemedicine. Cluster Computing pp 1–10 (2020)
20. Rhayma, H., Makhloufi, A., Hamam, H., Hamida, A.B.: Semi-fragile self-recovery watermarking scheme based on data representation through combination. Multimedia Tools Appl. **78**(10), 14067–14089 (2019)
21. Shehab, A., Elhoseny, M., Muhammad, K., Sangaiah, A.K., Yang, P., Huang, H., Hou, G.: Secure and robust fragile watermarking scheme for medical images. IEEE Access **6**, 10269–10278 (2018)
22. Singh, D., Singh, S.K.: Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability. J. Vis. Commun. Image Represent. **38**, 775–789 (2016)
23. Singh, P., Agarwal, S.: An efficient fragile watermarking scheme with multilevel tamper detection and recovery based on dynamic domain selection. Multimedia Tools Appl. **75**(14), 8165–8194 (2016)
24. Sreenivas, K., Kamakshiprasad, V.: Improved image tamper localisation using chaotic maps and self-recovery. J Vis Commun. Image Represent. **49**, 164–176 (2017)
25. Tai, W.L., Liao, Z.J.: Image self-recovery with watermark self-embedding. Signal Process. **65**, 11–25 (2018)
26. Thanh, T.M., Iwakiri, M.: Fragile watermarking with permutation code for content-leakage in digital rights management system. Multimedia Syst. **22**(5), 603–615 (2016)
27. Tsai, P., Hu, Y.C., Chang, C.C.: Novel image authentication scheme based on quadtree segmentation. Imaging Sci. J. **53**(3), 149–162 (2005)

28. Wang Z: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
29. Xie, X., Wang, C., Li, M.: A fragile watermark scheme for image recovery based on singular value decomposition, edge detection and median filter. Appl. Sci. **9**(15), 3020 (2019)
30. Yan, C., Gong, B., Wei, Y., Gao, Y.: Deep multi-view enhancement hashing for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020a)
31. Yan, C., Li, Z., Zhang, Y., Liu, Y., Ji, X., Zhang, Y.: Depth image denoising using nuclear norm and learning graph model. arXiv preprint (2020b) arXiv:200803741
32. Yan, C., Shao, B., Zhao, H., Ning, R., Zhang, Y., Xu, F.: 3d room layout estimation from a single rgb image. IEEE Transactions on Multimedia (2020c)
33. Yang, S., Qin, C., Qian, Z., Xu, B.: Tampering detection and content recovery for digital images using halftone mechanism. In: 2014 tenth International conference on intelligent information hiding and multimedia signal processing, pp 130–133 (2014)
34. Zhang, X., Wang, S.: Statistical fragile watermarking capable of locating individual tampered pixels. IEEE Signal Process. Lett. **14**(10), 727–730 (2007)

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com