The ISC Int'l Journal of Information Security

July 2022, Volume 14, Number 2 (pp. 147–156)
http://www.isecure-journal.org

Perfect Recovery of Small Tampers Using a Novel Fragile Watermarking Technique Based on Distributed Hamming Code

Faeze Rasouli [1], and Mohammad Taheri [1,*]
[1] *Computer Science and Engineering and Information Technology Department, Shiraz University, Shiraz, Iran.*

ARTICLE INFO.

Article history:
Received: May 6, 2021
Revised: August 6, 2021
Accepted: January 20, 2022
Published Online: January 22, 2022

Keywords:
Authentication Code, Fragile Watermarking, Hamming Code, Tamper Detection and Recovery

Type: Research Article

doi: 10.22042/ISECURE.2022.284952.670

dor: 20.1001.1.20082045.2022.14.2.3.5

ABSTRACT

Fragile watermarking is a technique of authenticating the originality of the media (e.g., image). Although the watermark is destroyed with any small modification (tamper), it may be used to recover the original image. There is no method yet, based on our knowledge, to guarantee the perfect recovery of small tampers. Although data-bits are embedded in Least Significant Bits of some other pixel(s), a tamper may destroy both data and authentication sets which makes recovery impossible. In this paper, a novel fragile watermarking scheme is proposed for both tamper detection and tampered image recovery. Here, all bits are reorganized in virtual pixels distributed in the image called as Distributed Pixels (DP). Distance of each pair of bits in a DP is sufficiently large. This is why; tampers smaller than a threshold, cannot destroy more than one bit of a DP. Hamming code guarantees that changing at most one bit can be perfectly detected and recovered. Then, Hamming(7,4) is extended to (8,5) to support embedding in eight-bits pixels. According to the experimental results, the proposed method could perfectly detect and recover the tampered parts not greater than a quarter of image in diameter. It also achieved acceptable performance in other conditions, compared to state-of-the-art methods.

1 Introduction

Information hiding typically includes the three disciplines: cryptography, watermarking, and steganography. Cryptography is encrypting a block of information to a block of data which is hardly invertible unless associated key is provided. It can be used for encoding a message, message authentication or author authentication (signing). Although, message encryption leads to have security to some extent, but it may practically attract the attackers' attention. This is why; researches have recently been interested in steganography to hide the message in some media called as cover. Watermarking however, is an alternative for data or owner (creator) authentication. In steganography, the message has the most important content and the cover may be modified in order to hide it. But watermarking generates a proper block of data embedded (hidden or observable) in the cover media to authenticate the owner or originality of the cover. In the former case, called robust watermarking, the watermark, as a representation of the owner's properties, is embedded in the cover and is resistant against image distortion [1, 2]. In the latter case, called as fragile watermarking, a small modification in the cover (denoted by tamper) leads to destroying the watermark that means the cover is not original more [3–16]. Semi-fragile watermarking is also pro-

* Corresponding author.
Email addresses: f.rasouli@shirazu.ac.ir, motaheri@shirazu.ac.ir
ISSN: 2008-2045 © 2020 ISC. All rights reserved.

posed in some researches [17].

The evolution and the advancement of digital technology, and at the same time, developing the communication tools, such as Internet, made communication easier to transmit digital data with minimal error. As a result, multimedia data can be easily published, edited, and copied. Free access and exchange of information increase the possibility of data misuse as well. Several watermarking techniques have been proposed in the literature to increase the security of multimedia information (including digital images). Semi-fragile and fragile watermarking approaches are frequently applied to tamper detection. In this paper, grayscale images are considered as the covering media.

One of the most frequent types of tampers affects spatial view of the images (by changing the value of some pixels in the image). This is why; many researchers have been recently interested in spatial fragile watermarking. Most of these techniques are initially applied on grayscale images whereas, each pixel value is represented by eight bits. The main goal of these approaches is usually coding the Most Significant Bits (MSB) of the image in the Least Significant Bits (LSB). In other words, due to preserve the image quality, only LSBs may be modified by new bits. The new bits, called as authentication bits, store some information about MSBs, called data-bits, to detect any change of them. Recent researches store authentication bits far from associated data-bits. Hence, if a tamper changes some data-bits, authentication ones are probably unchanged and can be used to detect the tamper and may even recover the original values.

Another attractive topic in fragile watermarking is image recovery. This task is using information of watermark in order to reconstruct the image. In other words, assuming the watermark associated with the tampered part of the image itself is not modified, the goal is finding a set of pixels consistent with both watermark and other parts of the image. Therefore, if both pixels and associated watermarks are changed, recovery is impractical.

Due to have acceptable image quality, number of authentications bits are usually less than number of data-bits. This is why; the perfect recovery is not guaranteed in the literature even with especial conditions. In this paper, Hamming code is used to generate authentication bits from data ones. Based on this technique, undesired changes can be perfectly recovered if just one of the bits (data or authentication) is changed. In order to achieve this property, all the bits involved in a Hamming coding are distributed in the image to be sufficiently far. Hence, if the tamper size is less than a threshold, it is guaranteed that, at most one of the bits, used in or generated by a Hamming code, is modified. Therefore, the perfect recovery is guaranteed in the case of having sufficiently small tampers.

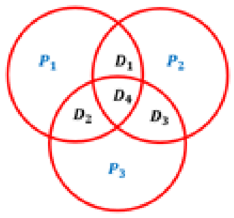


Figure 1. The relationship between data-bits and parity check bits

In the rest of the paper, Hamming code technique is briefly described in Section 2. Section 3 is dedicated to related work. The proposed novel fragile watermark scheme, including watermark embedding, the tampered area's detection, and recovery of the tampered area, is explained in Section 4. Experimental results are presented in Section 5. Finally, the paper is concluded in Section 6.

2 Hamming Code

Hamming code was developed by R.W. Hamming for error detection and correction [18]. Hamming(7,4) is an ordered set of seven bits where, three of them are called parity check bits and are generated by a linear function of other four bits called as data-bits. In other words, the bits are represented by $[\overrightarrow{D}, \overrightarrow{P}]$ where, $\overrightarrow{D} = [D_1, \ldots, D_4]$ and $\overrightarrow{P} = [P_1, P_2, P_3]$ are data and parity bits, respectively. In addition, $\overrightarrow{P} = \overrightarrow{D}M$ as a linear function of $\overrightarrow{D}$ (all vectors and matrix M contain binary values and operation are done in F_2). Hamming(7,4) is capable to detect up to two simultaneous bit errors and correcting single-bit errors. Suppose D_1, D_2, D_3 and D_4 are data-bits, parity bits P_1, P_2 and P_3 are calculated by XOR operation of associated data-bits. The relationships between the check bits and associated data-bits are shown in (1) where, $\oplus$ is an exclusive OR (XOR) operation. Also, this relationship is illustrated in Figure 1.

$$\begin{aligned} P_1 &= D_1 \oplus D_2 \oplus D_4 \\ P_2 &= D_1 \oplus D_3 \oplus D_4 \\ P_3 &= D_2 \oplus D_3 \oplus D_4 \end{aligned} \tag{1}$$

As shown in Figure 1, the intersection of all three circles is D_4. This means that D_4 participates in all three circles to generate the corresponding check bits. The Hamming decoder is responsible for generating the Syndrome vector $S = [S_1 S_2 S_3]$ from received vector $[D_1, D_2, D_3, D_4, P_1, P_2, P_3]$ that represents which parity-check equations are not satisfied. The Syndrome vector is calculated using the following statement.

$$\begin{aligned} S_1 &= D_1 \oplus D_2 \oplus D_4 \oplus P_1 \\ S_2 &= D_1 \oplus D_3 \oplus D_4 \oplus P_2 \\ S_3 &= D_2 \oplus D_3 \oplus D_4 \oplus P_3 \end{aligned} \quad (2)$$

If the Syndrome vector is equal to [0 0 0], it means that no error has occurred. Otherwise, it exactly indicates which bit of the received vector is in error. To better understand, suppose that the Syndrome vector is equal to [1 0 1], which indicates that 1^{st} and 3^{rd} equations are not consistent. The only bit which can toggle the consistency of just these equations is D_2 as the only possible error on a single bit.

3 Related Work

Various researchers on fragile watermarking have proposed several algorithms. The first study to be described is the watermarking for tamper detection and recovery proposed by Lin, *et al.* [3] as the base of most of spatial watermarking methods. In this method, the original image was divided into non-overlapping blocks of size 4×4 pixels. Each block is divided into four sub-blocks of size 2×2 pixels. Six MSBs and two LSBs of each pixel are data and watermark bits, respectively. Consequently, each sub-block has 2bits $\times$ 4pixels = 8bits to store the watermark. Six bits of each watermark is used for image recovery, resulting from the average intensity of six MSBs of associated sub-block. The other two bits are the parity check bits to authenticate the watermark. In the rest of this section, two sets of other related works have been considered: distributed watermarking and watermarking based on Hamming code.

3.1 Distributed Watermarking

Distributed watermarking is used in this section to address the methods which store most significant information of a block in more than one location in the image. A dual watermark scheme for image tamper detection and recovery was presented by Lee and Shinfeng [4]. Their proposed method improved Lin's method by solving main drawback. The difference is that, a second chance is provided to recover the destroyed block in such a way that, for each non-overlapping block, two copies of the watermark are maintained. This novelty decrease the chance of distortion of both data and watermark-bits of a block. The size of each block is 2×2 pixels with three watermark-bits as LSBs of each pixel. Hence, the watermark length is 3bits $\times$ 4pixels = 12bits. Ten bits to store average intensity of two other blocks respect to five MSBs and two remained bits are for watermark parity check.

In 2020, Sarkar, *et al.* proposed two schemes for image tamper detection and restoration [5], one of which works in the spatial domain and implements a quadruple watermarking approach. In their proposed method, four chance is provided to recover the destroyed block. The watermark is generated from four blocks and is embedded in other identically four blocks using the mapping algorithm. The size of each block is 3×3 pixels. The watermark length is 18 bits and the average intensity of each block (considering just four MSBs of each pixel) is used to generate the recovery bits. Finally, the watermark, with two parity check bits, is embedded in the two LSBs of nine pixels of the target blocks. Although just two bits are reserved for watermark, but due to having many copies of watermarks, just four MSBs of destroyed blocks can be recovered.

Although both of these methods distribute the watermarks, but one complete copy of each watermark is stored in each location. Having more than one copies, leads to decreasing the embedding rate and image quality or recovery power. In addition, in both of them, average intensity is stored which cannot perfectly recover the destroyed parts. In this paper, however, just one copy of the watermark is stored but its pixels are distributed in the image to be sufficiently far. It leads to increase the chance of changing the watermark but just one bit per watermark if the tamper is sufficiently small. Then, it is tried to recover this bit by Hamming recovery.

3.2 Watermarking Based on Hamming Code

In 2007, Chan proposed an image authentication method using the Hamming code technique [7]. In that method, just Hamming code is used for generating three watermark-bits from four data-bits. One of the drawbacks of this method was that the most significant bit per pixel was predicted based on neighboring pixels. Since the MSB plays an important role in the intensity value of one pixel, the recovery error may be high on edges. Chan reduced the effect of this error in [6] by determining the most-significant bit of each pixel according to its check bits.

In any tampered pixel, probability of toggling more than one out of three watermark-bits is 0.5, and probability of more than one out of four data-bits is 11/16. Hence, most of the times, the recovery property of Hamming code cannot be used in these methods.

In comparison with the related works presented in this section, the proposed method distributes each watermark without any duplication which leads to more effectiveness in recovery. Although, the chance of watermark modification increases, but at most one bit of data or associated watermark may be changed for sufficient small tampers. This single toggled bit can be detected and recovered based on Hamming code applied on embedding the distributed pixel (data

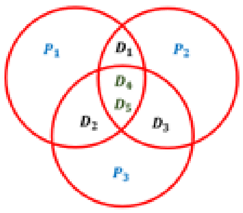


Figure 2. The relationship between data-bits and parity check bits in Hamming(8,5)

and watermark). As another difference, the original value of each pixel not averaged intensity of a block can be discovered in the proposed method. Hence, perfect recovery is the main achievement of this paper in the case of having tampers smaller than a specified threshold.

4 Proposed Method

In this paper, a new method is presented that takes the Hamming code into consideration for tamper detection and tamper recovery. This approach consists of three major procedures: embedding, detection and recovery procedures. The embedding procedure describes how the watermark is embedded into each pixel's LSBs in the original image. The tamper detection procedure aims to localize the tampered areas, and finally, the recovery procedure is an attempt to recover the areas that have been tampered. The details of the proposed schemes are described as follows.

4.1 The Embedding Procedure

Commonly, the grayscale intensity is stored as an 8-bit integer giving 256 possible different shades of gray from black to white. In this paper, three LSBs of pixels are modified to embed information of five MSBs. As mentioned in Section 2, Hamming(7,4) encodes four bits of data into seven bits by adding three parity bits. By replacing 3 LSBs with parity check bits, Hamming can be used as embedding algorithm for watermarking. However, as mentioned above, there are five bits to be embedded in three ones and a Hamming(8,5) is required.

If information from the fifth most significant bit is not recorded in the watermarked image, the image recovery will be weakened. To solve this problem, a modified Hamming is proposed in (3).

$$\begin{aligned} P_1 &= D_1 \oplus D_2 \oplus D_4 \oplus D_5 \\ P_2 &= D_1 \oplus D_3 \oplus D_4 \oplus D_5 \\ P_3 &= D_2 \oplus D_3 \oplus D_4 \oplus D_5 \end{aligned} \tag{3}$$

Indeed Hamming(7,4) has been extended to Hamming(8,5) in this paper, as shown in Figure 2. As mentioned in Section 2, one of data-bits (D_4) participates in generating all three check bits. For Hamming(8,5) both of D_4 and D_5 play this role. In other words, $D_4 \oplus D_5$ in Hamming(8,5) is used instead of just D_4 in Hamming(7,4). In this case, if just one bit is modified, it can be exactly discovered or it will be understood that one of D_4 or D_5 is toggled (50% error in this case). The next challenge is proposing an embedding method to decrease the probability of modifying more than one bit involved in a Hamming code.

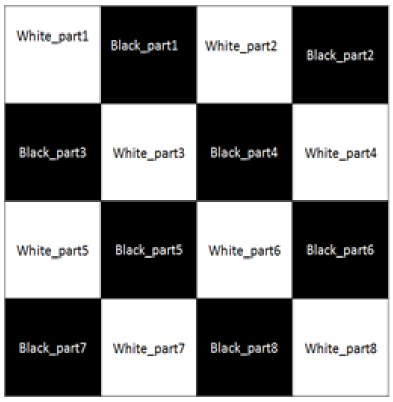


Figure 3. The arrangement of the white and black parts next to each other

First, the grayscale image is divided into sixteen independent parts: eight white and eight black parts. The parts are divided in such a way that two parts of the same color have no common edge, exactly like a chessboard depicted in Figure 3. Each color is independently embedded in itself. All parts have the same shape and size. Having parts of size k×k, there are k^2 pixels in a part. Each pixel of a white part is associated with seven other pixels in the same location of other white parts. Hence there are k^2 tuples of white pixels whereas each tuple is a set of eight pixels similarly located in the eight white parts (similarly for the black color). A tuple is depicted in Figure 4 for both white and black colors. As can be seen in Figure 4, distance of each pixel from co-tuple pixels is at least $\sqrt{2}k$. Therefore, if the diameter of the tamper (maximum distance of the tampered pixels) is less than $\sqrt{2}k$, it is guaranteed that at most one of the pixels per tuple is modified. The embedding process is executed in four rounds. The first and second rounds are dedicated to White and the next ones are corresponding to the Blacks.

In each round, four parts of one color are considered to be embedded in other parts of that color. For example, in the first round, suppose the four upper white parts of the image, are selected to be embedded in the lower white parts. In this case, for each white tuple of pixels, four pixels in the upper half of image are selected to provide data-bits D_1, D_2, D_3 and D_4 from their four MSBs (one bit per pixel). Another pixel of the tuple (from the lower part) is also selected

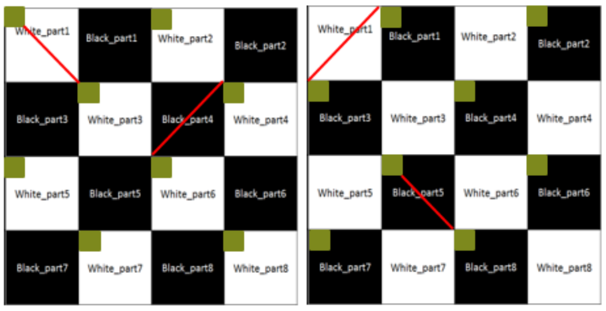


Figure 4. Distance between the bits of Hamming code distributed in (a) white parts, (b) black parts

to provide D_5 by its fifth most significant bit. These five bits are coded by Hamming(8,5) to produce three check bits P_1, P_2 and P_3 as presented in (3). The check bits are then embedded in three LSBs of other three pixels in the tuple (one bit per pixel). This process is done in four iterations.

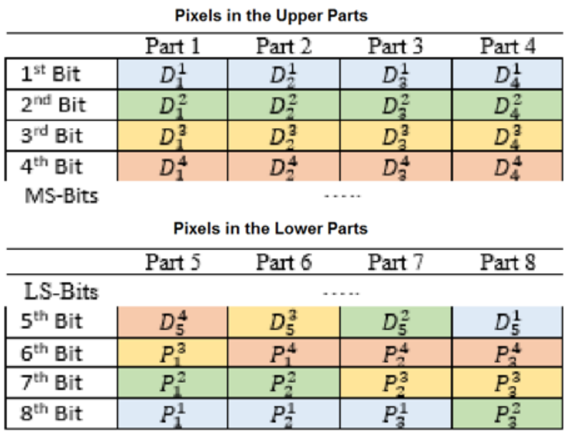
Each upper pixel of the tuple provides one of its four MSBs in each iteration. Each lower pixel, once provides D_5 and three times embed a parity check bit in one of its three LSBs. After four iterations, all four MSBs of the upper white parts and fifth MSBs of all lower white parts are embedded in all LSBs of lower white parts. This process is inversely repeated in the second round for embedding MSBs of lower parts in upper ones using fifth MSBs in the upper parts. Also, these rounds are repeated for black parts. Totally, eight Hamming codes are used for each tuple of pixels independently.

Follow Figure 5 to find out the data-bits and the check-bits' locations in the first round (also can be used for the third round). In this figure, Hamming code on a single tuple of pixels is presented. Four MSBs of pixels in the upper parts and Four LSBs pixels in the lower parts have been shown (the 1^{st} bit is the most significant). The bits with the same color are used for embedding in a same iteration where, D_j^i represents D_j of Equation (3) in i^{th} iteration.

4.2 The Tamper Detection Procedure

Tamper detection is conducted at two levels. At the first level, the image is divided into sixteen independent parts, exactly as in the embedding phase. To localize the tampered area, the Syndrome vector is first generated for each tuple as described in Section 2. The bits and the relationship between them to generate the Syndrome vector must be exactly the same as that used in the embedded step.

If the Syndrome vector is equal to [000], no error has occurred; otherwise, the Syndrome vector's value, indicates the bit in which the error occurred. If there



Pixels in the Upper Parts

	Part 1	Part 2	Part 3	Part 4
1st Bit	D_1^1	D_2^1	D_3^1	D_4^1
2nd Bit	D_1^2	D_2^2	D_3^2	D_4^2
3rd Bit	D_1^3	D_2^3	D_3^3	D_4^3
4th Bit	D_1^4	D_2^4	D_3^4	D_4^4
MS-Bits		-----		

Pixels in the Lower Parts

	Part 5	Part 6	Part 7	Part 8
LS-Bits		-----		
5th Bit	D_5^4	D_5^3	D_5^2	D_5^1
6th Bit	P_1^3	P_1^4	P_2^4	P_3^4
7th Bit	P_1^2	P_2^2	P_2^3	P_3^3
8th Bit	P_1^1	P_2^1	P_3^1	P_3^2

Figure 5. Four MS-Bits and LS-Bits of co-tuple pixels in upper and lower parts, respectively during the first round of embedding

is any error bit (bit which is inconsistent based on Hamming) in the pixel, that pixel is marked as the tampered pixel. As shown in the Figure 6, pixels with one or more error-bits (where, the error-bit is indicated by e) are considered as tempered pixels (indicated by T-pixel) and displayed in red.

This procedure is done for the whole image to localize the tampered area. It is important to note that when the Syndrome vector is equal to [1 1 1], there are two candidate bits to consider the error event in them. The reason is that the fifth bit was added as the data-bit in the Hamming(8,5) structure. Therefore, no decision is made on the error-bit and the decision is postponed to the next level.

In the second level, the state that Syndrome vector is equal to [1 1 1] is checked. In this case, the candidate bit of error which has a larger number of tampered pixels in the neighborhood of 3 × 3 pixels around it, is selected as the error-bit. To better understand, Figure 7 shows some tampered pixels that have been marked and are available from the first level. Suppose two bits in pixels A and B are two candidates of error-bits. The bit in pixel A is considered as the error-bit, because pixel A has more tampered pixel around 3 × 3 pixels of neighborhood than pixel B. Then pixel A is also marked as tampered pixel.

4.3 The Recovery Procedure

In the recovery phase, the Syndrome vector is required to indicate the tampered bit, as mentioned earlier. The vectors [0 0 0] indicates no error. For other Syndrome vectors except of [1 1 1] the tampered bit is determined and is toggled according to Syndrome vector's value. For the case of Syndrome [1 1 1], there are two candidate bits for tampered bits. In this case, the one whose pixel in the detection phase has been

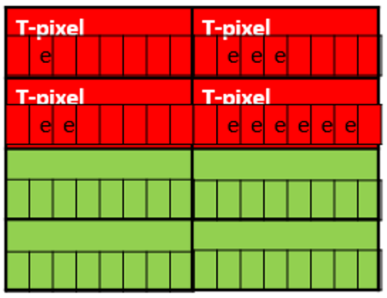


Figure 6. Tampered pixels and error bits

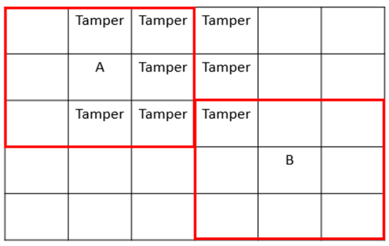


Figure 7. 3 $\times$ 3 Neighbor pixels and error bit decision

marked as a tampered pixel is toggled.

The result of this phase is a very high-quality recovered image. As mentioned, for the tampers with diameter less than $\sqrt{2}k$ where, each part is of size k×k, there is at most one bit-modification in each tuple of pixels. As can be seen in Figure 4, the size of image is 4k×4k. Since, if diameter of the tamper is not greater than a quarter of the diameter of the image, each tuple has at most one modified pixel.

Since, Hamming is applied on each tuple separately, the recovery can be done perfectly except for the case of changing D_4 or D_5. In this case, the ambiguity can be solved if there is a witness to mark one of corresponding pixels in the tuple as the tampered pixel. Due to that, eight different Hamming codes are used for each tuple, there is a high probability of occurring cases in which the tampered pixel can be deterministically identified. As mentioned in the tamper detection phase, in order to have more accuracy, the marked pixels in the neighborhood of each pixel are also counted for reasoning.

As mentioned, the distributed Hamming code in the white and black parts is applied independently. As a consequence, if only one white part and one black part have been tempered (which has a diameter greater than the threshold), the image can be also recovered perfectly. Figure 8 shows some different tamper forms, including a maximum of one white part and one black part, in which case, the image will be fully recovered. In this case, out of the sixteen parts available, tampering on two parts (12.5% of the image) can be fully recovered.

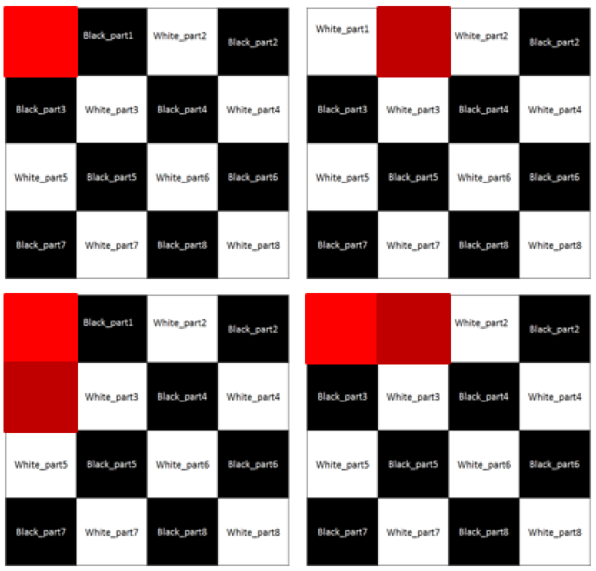


Figure 8. Some temper forms with full recovery

5 Experimental Results

In this section, some experimental results are given which indicate the validity of the proposed scheme in both tamper detection and recovery. For quantitative evaluation, peak signal-to-noise ratio (PSNR), was introduced to evaluate the PSNR of image I_1 relative to image I_2. PSNR is defined in (4) and (5).

$$PSNR = 20 \times Log_{10}(\frac{255}{\sqrt{MSE}}) \tag{4}$$

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |x(i,j) - y(i,j)|^2 \tag{5}$$

The symbols x(i ,j) and y(i ,j) represent the pixel values at the position (i,j) in the images x and y, respectively. Also, M and N represent the pixel numbers for the width and the height of the image, respectively. If x and y are the original and watermarked images, regardless of the order, PSNR evaluates the quality of embedding procedure. If these images are watermarked and recovered images, PSNR is a metric of recovery strength. In addition, Structural Similarity Index Measure (SSIM) has been recently proposed to measure the similarity of an image with its original version based on structure of the image not bit errors as shown in (6).

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{6}$$

Where, μ_x and σ_x^2 represent the mean and variance of pixel values in the image x, respectively. Moreover, σ_{xy} is the covariance of pixel values of the images x and y. Finally, c_1 and c_2 are constant values to prevent weak denominator and are set to default values

Figure 9. Some most widely used standard test images (Original Images): “Lena”, “Cameraman”, “Pepper” and “Barbara”

presented in [19]. As can be seen, SSIM compares total similarity of the images respect to mean, variance and covariance of pixel values and it has less concentration on specific tampered pixels in comparison with PSNR. This problem along with having no SSIM for most of experiments of related works, leads to selecting PSNR as the main evaluation metric in this paper.

In the current investigation, some standard images were used for the experiments. Also, the four most widely used standard images have been illustrated in Figure 9, which are “Lena”, “Cameraman”, “Pepper” and “Barbara”. All these grayscale images are of size 512 × 512 pixels. The watermarked image is the output of the embedding step. If a proper embedding is done, the watermarked image has high quality; in other words, the original image cannot be distinguished visually from the watermarked image, as shown in Figure 10.

As mentioned, embedding PSNR measures imperceptibility and similarity between the original and the watermarked images. Table 1 demonstrates the PSNR of watermarked images for 10 standard images including the ones presented in Figure 9 and Figure 10. The proposed PSNR could be better than 37.9 as the expected embedding PSNR of a watermarking method with embedding rate of 3 pixels on a random image.

As previously mentioned, if the tamper includes a maximum of two parts from different colors, the image can be recovered perfectly. This scenario is called as scenario #1 in which, the tampered area is limited and does not exceed the area of two adjacent parts (in size). In scenario #2, there is no limitation, and the tampered area can be a square environment, that its width is larger than the width of a single part. Figure 11-15 show the result of the recovered image relative to various tampered sizes and scenarios. Also, Table 2 demonstrates the recovery PSNR scenarios #1 and #2 with tamper size 10% on more images.

Table 1. PSNR of the watermarked image relative to the original image

Original image	PSNR
pepper	38.08
Lena	38.06
Barbara	38.08
Cameraman	38.09
Plane	38.12
Baboon	38.08
Boat	38.05
Zelda	38.08
Elaine	38.07
Home	38.05

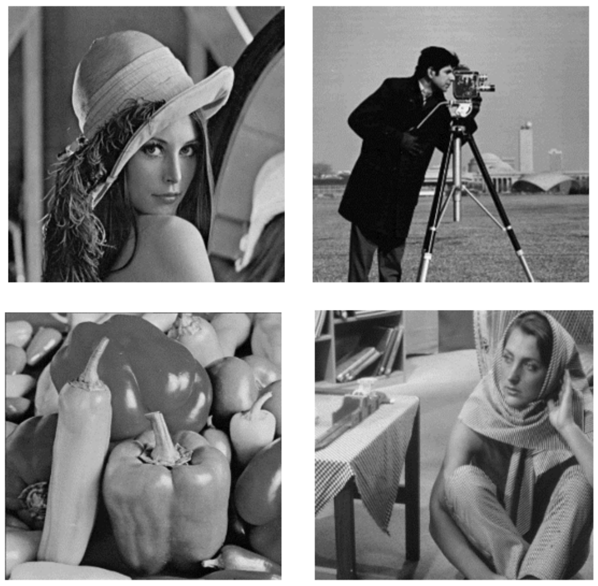


Figure 10. Watermarked Images of Figure 9

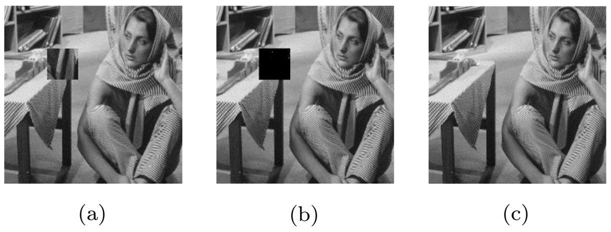


Figure 11. Image of Barbara: (a) 3% tampered with scenario #1, (b) the detected tampered regions, (c) recovered image

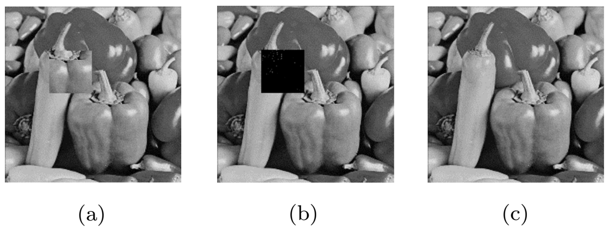


Figure 12. Image of Pepper: (a) 6% tampered with scenario #1, (b) the detected tampered regions, (c) recovered image

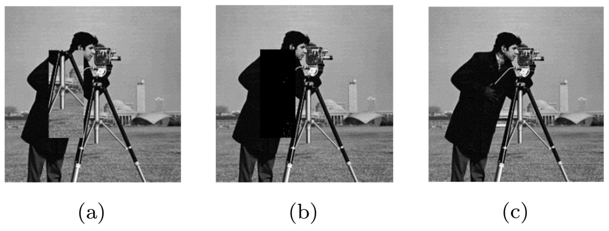


Figure 13. Image of Cammeraman: (a) 10% tampered with scenario #1, (b) the detected tampered regions, (c) recovered image

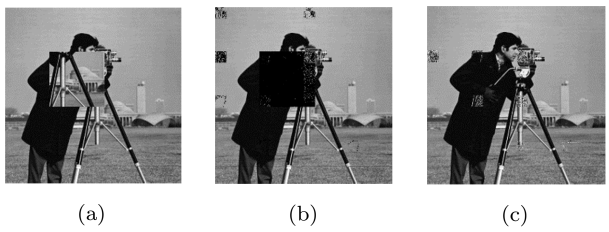


Figure 14. Image of Cammeraman: (a) 10% tampered with scenario #2, (b) the detected tampered regions, (c) recovered image

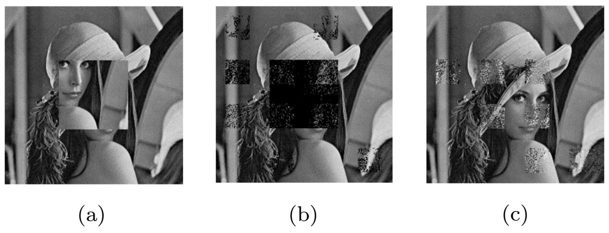


Figure 15. Image of Lena: (a) 15% tampered with scenario #2, (b) the detected tampered regions, (c) recovered image

As shown in Table 2, the proposed method achieves a perfect recovery on scenario #1 with PSNR of ∞. For scenario #2 however, there are some errors in recovery. This is why; both PSNR and SSIM have been reported in this case. SSIM have reported in specific conditions in a few researches. Although, the conditions are not comparable, the achieved SSIM of the proposed method is acceptable, more or less. In Table 3 and Table 4, PSNR of the recovered images are reported to further demonstrate the performance of the proposed scheme in comparison with the state-of-the-art methods. Table 3 is dedicated to Hamming Code based watermarking methods presented in [6] and other related works have been compared in Table 4. Table 3 shows the PSNR of the recovered image of Chan and Chi-Shiang's [6] proposed scheme. This method [6] is one of the few articles in which, in addition to using Hamming Code, some recovery methods were proposed and associated results were reported. In their experiments, the image Lena with size 512 × 512, and the tamper size is 64 × 64. The embedding rate indicates the average number of bits used to embed the authentication data. The higher value of the embedding rate, the lower value of PSNR in the embedding stage. The result of method#4 recovers the image perfectly, but in the embedding phase, PSNR value decreases. In contrast, the proposed method maintains maximum PSNR value with three watermark-bits per pixel.

Table 2. PSNR of the recovered images relative to the watermarked ones

image	Scenario #1	Scenario #2	
	PSNR	PSNR	SSIM
Pepper	∞	49.32	0.96
Lena	∞	49.09	0.96
Barbara	∞	48.87	0.95
Cameraman	∞	48.91	0.96
Plane	∞	49.46	0.96
Baboon	∞	49.09	0.96
Boat	∞	49.38	0.96
Zelda	∞	48.45	0.95
Elaine	∞	48.58	0.95
Home	∞	48.50	0.95

Table 4 demonstrates the results of the proposed method compared to other state-of-the-art methods with different tamper sizes. As can be seen, for tamper size of 10%, the proposed method significantly outperforms others. However, by increasing the tamper size, the condition of having at most one bit-modification in the tuple is not held. This is why; for higher tamper sizes, the performance of the proposed method degrades. In Table 4, the Lena image of size 512 × 512 was used for the experiments. In order to measure the efficiency of the proposed method and state-of-the-art methods compared in this paper, it should be mentioned that all of them have a constant time-complexity for each pixel in all embedding, tamper detection and recovery phases. In other words, the total time complexity of these methods is linearly related to the number of pixels.

6 Conclusion

In this paper, a novel fragile watermarking method has been proposed to detect tamper area in the image

Table 3. PSNR of the recovered image in comparison with hamming code based watermarking methods

Paper	Embedding rate	PSNR (dB) of Recovered image
Ref [6] method #1	3	51.97
Ref [6] method #2	3.25	52.90
Ref [6] method #3	3.49	53.42
Ref [6] method #4	4	∞
Proposed method	3	∞

Table 4. PSNR of the recovered image relative to the tampered size

Paper	10% Tampered	20% Tampered	25% Tampered
Ref [4]	35.17	–	33.45
Ref [5](Quad method)	41.10	39.45	–
Ref [8]	45.85	–	–
Ref [9]	37.50	–	33.95
Ref [10]	38.69	37.15	–
Ref [11]	45.09	40.58	39.50
Ref [12]	44.15	41.83	–
Ref [13]	48.21	45.07	–
Ref [14]	40.48	36.57	34.80
Ref [15]	47.00	43.00	41.00
Ref [16]	36.00	30.00	29.00
Proposed method (Scenario #1)	∞	39.44	37.16
Proposed method (Scenario #2)	49.09	39.44	37.16

and recover the tampered areas based on Hamming code. In this paper, first Hamming code (7,4) has been extended to (8,5). Then, encoding of Hamming(8,5) has been applied to generate authentication code for distributed pixels. Also decoding Hamming(8,5) has been used to detect and recover tampered pixels. Theoretically and based on experimental results, if the tamper's diameter is not greater than a quarter of the image diameter; the proposed method recovers the image perfectly. Also, in greater sizes of temper, results are comparable to some other state-of-the-art methods. Extending the work to support more than one bit modification per distributed pixel, improving the embedding PSNR, increasing the upper bound of the tamper for perfect reconstruction and extending the proposed method for two watermark-bits embedding may be considered in the future.

References

[1] Satendra Pal Singh and Gaurav Bhatnagar. A new robust watermarking system in integer dct domain. *Journal of Visual Communication and Image Representation*, 53:86–101, 2018.

[2] Xiaobing Kang, Yajun Chen, Fan Zhao, and Guangfeng Lin. Multi-dimensional particle swarm optimization for robust blind image watermarking using intertwining logistic map and hybrid domain. *Soft Computing*, 24(14):10561–10584, 2020.

[3] Phen Lan Lin, Chung-Kai Hsieh, and Po-Whei Huang. A hierarchical digital watermarking method for image tamper detection and recovery. *Pattern recognition*, 38(12):2519–2529, 2005.

[4] Tien-You Lee and Shinfeng D Lin. Dual watermark for image tamper detection and recovery. *Pattern recognition*, 41(11):3497–3506, 2008.

[5] Dipabali Sarkar, Sarbani Palit, Sukalyan Som, and KN Dey. Large scale image tamper detection and restoration. *Multimedia Tools & Applications*, 79, 2020.

[6] Chi-Shiang Chan. An image authentication method by applying hamming code on rearranged bits. *Pattern Recognition Letters*, 32(14):1679–1690, 2011.

[7] Chi-Shiang Chan and Chin-Chen Chang. An efficient image authentication method based on hamming code. *Pattern Recognition*, 40(2):681–690, 2007.

[8] Surya Bhagavan Chaluvadi and Munaga VNK Prasad. Efficient image tamper detection and recovery technique using dual watermark. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pages 993–998. IEEE, 2009.

[9] Faranak Tohidi and Manoranjan Paul. A new image watermarking scheme for efficient tamper detection, localization and recovery. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 19–24. IEEE, 2019.

[10] Irshad Ahmad Ansari, Millie Pant, and Chang Wook Ahn. Svd based fragile watermarking scheme for tamper localization and self-recovery. *International Journal of Machine Learning and Cybernetics*, 7(6):1225–1239, 2016.

[11] Durgesh Singh and Sanjay K Singh. Dct based efficient fragile watermarking scheme for image authentication and restoration. *Multimedia Tools and Applications*, 76(1):953–977, 2017.

[12] Behrouz Bolourian Haghighi, Amir Hossein Taherinia, and Amir Hossein Mohajerzadeh. Trlg: Fragile blind quad watermarking for image tamper detection and recovery by providing compact

digests with optimized quality using lwt and ga. *Information Sciences*, 486:204–230, 2019.

[13] Navid Daneshmandpour, Habibollah Danyali, and Mohammad Sadegh Helfroush. Image tamper detection and multi-scale self-recovery using reference embedding with multi-rate data protection. *China Communications*, 16(11):154–166, 2019.

[14] Vishal Rajput and Irshad Ahmad Ansari. Image tamper detection and self-recovery using multiple median watermarking. *Multimedia Tools and Applications*, 79(47):35519–35535, 2020.

[15] Assem Abdelhakim, Hassan I Saleh, and Mai Abdelhakim. Fragile watermarking for image tamper detection and localization with effective recovery capability using k-means clustering. *Multimedia Tools and Applications*, 78(22):32523–32563, 2019.

[16] Omer Hemida and Hongjie He. A self-recovery watermarking scheme based on block truncation coding and quantum chaos map. *Multimedia Tools & Applications*, 79, 2020.

[17] Imran Sikder, Pranab Kumar Dhar, and Tetsuya Shimamura. A semi-fragile watermarking method using slant transform and lu decomposition for image authentication. In *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 881–885. IEEE, 2017.

[18] Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

[19] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

Faeze Rasouli received her B.Sc. degrees in hardware engineering from Hamedan University of Technology, Hamedan, Iran, in 2017. Now, she is a M.Sc. student in the Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran. Her research interests are database aecurity, image processing, computer vision, machine learning.

Mohammad Taheri was born in 1983. He achieved B.Sc., M.Sc. and Ph.D. degrees as an outstanding student in Computer Science & Engineering department of Shiraz University (Iran) since 2001 until 2013. He started his job, as the faculty member (2014) in that department, with researches in machine learning, fuzzy systems, large margin classifiers, optimization, modeling and information hiding.