The ISC Int'l Journal of Information Security

July 2023, Volume 15, Number 2 (pp. 230–239)

http://www.isecure-journal.org

A Fragile Watermarking by Hamming Code on Distributed Pixels with Perfect Recovery for Small Tampers **

Faeze Rasouli [1], Mohammad Taheri [1,*], and Reza Rohani Sarvestani [2]

[1] *Department of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran*

[2] *Faculty of Technology and Engineering, Shahrekord University, Shahrekord, Iran*

ARTICLE INFO.

Article history:

Received: December 25, 2021

Revised: May 6, 2022

Accepted: July 1, 2023

Published Online: July 5, 2023

Keywords:

Distributed Pixel, Fragile Watermarking, Hamming Code, Image Reconstruction, Tamper Detection

Type: Research Article

doi: 10.22042/isecure.2023.321411.740

dor: 20.1001.1.20082045.2023.15.2.6.5

ABSTRACT

Fragile watermarking is embedding a watermark in a media (an image in this paper) such that even small changes, called tamper, can be detected or even recovered to prevent unauthorized alteration. A well-known category of spatial fragile watermarking methods is based on embedding the watermark in the least significant bits of the image to preserve the quality. In addition, Hamming code is a coding algorithm in communication that transmits the data bits by augmenting some check bits to detect and recover single-bit modifications precisely. This property was previously used to detect and perfectly recover the images modified by small tampers less than a quarter of the image in diameter. To achieve this goal, the Hamming code is applied on a distributed pixel, bits of which are gathered from sufficient far pixels in the image. It guarantees that such tampers can toggle at most one bit of each distributed Hamming code that is recoverable. It was the only guaranteed perfect reconstruction method of small tampers, based on our knowledge. In this paper, the method has been extended to support distortion in two bits of a Hamming code by the use of common structures of distributed codes. It guarantees the recovery of tampers less than half of the image in width and height. According to the experimental results, the proposed method achieved better performance, in terms of recovering the tampered areas, in comparison to state-of-the-art.

1 Introduction

In recent years, with the rapid development of computer technology and the Internet, multimedia information in a digital format especially digital images, has become more and more popular. Also, much digital content (image, audio, and video) can be frequently published, copied, and edited. Image processing tools, like Photoshop, are rapidly developing. These tools allow users to modify images easily, imperceptibly, and in the shortest possible time. As a result, image security has become more important as a major issue. Watermarking is one of the most effective and powerful methods to increase the security of digital images. Based on the functionality, watermarking is divided into robust, semi-fragile, and fragile methods. Robust watermarking [1, 2] mainly aims at preventing unauthorized removal or intentional distortion of ownership marks. The embedded watermark in a robust technique must be very resistant to all kinds

* Corresponding author.

**This article is an extended/revised version of an ISCISC'21 paper.

Email addresses: f.rasouli@shirazu.ac.ir, motaheri@shirazu.ac.ir, rrohani.cse@gmail.com

ISSN: 2008-2045

of attacks, such as lossy compression, filtering, and geometric scaling. Primarily, robust techniques are designed for copyright protection of multimedia data. Semi-fragile watermarking [3] schemes are sensitive to non-intentional manipulations such as cutting, rotation, and shifting. These methods are, however, tolerant to allowable modifications. In contrast with robust watermarking, the embedded watermark in fragile watermarking [4, 5] will be destroyed when the digital content is modified. It is the main characteristic of fragile watermarks for image authentication. Image can be considered as a matrix including a pixel as each one of its elements. Grayscale images have only a single channel, and each pixel consists of eight bits giving 2^8=256 possible different shades of gray from black to white. Several algorithms have been proposed by various researchers for fragile watermarking [6]. Considering spatial approaches, some works are based on pixel replacement. For example, in [7], the pixel values are replaced by 54 selected values considering a prime distribution in the range [0-255]. However, a common approach, in order to preserve the quality of the tampered image, is embedding the information of the most significant bits in the least significant ones [8]. There are some models which try to simulate watermarking as a source-channel coding problem. For example, in [9], a fragile watermarking is proposed by hashing the compressed image to detect the tampers, and then modified bits are recovered by a dynamic programming method inspired by concepts of data communication theory. Similarly, Hamming code as a communication coding algorithm is used in this paper but in order to perfect recovery by distributing the code in the image. In this paper, a fragile watermarking scheme for tamper detection and recovery is proposed. The Hamming code technique is used to detect and correct modified bits. In this paper, with a slight change in Hamming code algorithm, a modified algorithm has been introduced that can recover five bits of data using three parity bits. Also, parity bits and data are embedded in different pixels of the image so that if the image is damaged, the parity bits may be used for recovery. The rest of the paper is organized as follows. In Section 2, a brief description of Hamming code technique is given. Related work is expressed in Section 3. The proposed method is presented in detail in Section 4. Experimental results are given in Section 5. Finally, the conclusion is given in Section 6.

2 Preliminary

Hamming code was developed by R.W. Hamming for error detection and correction [10]. Hamming (7,4) is an ordered set of seven bits where, three of them are called parity check bits and are generated by a modular linear function of the other four bits, called data bits. Assume data bits are represented by D_1, D_2, D_3 and D_4. Then, three XOR functions of data bits generate the check bits P_1, P_2, and P_3. These functions are presented in Equation 1 where, XOR operation is denoted by $\oplus$.

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

The Hamming code is designed such that, if the check bits are not consistent with the data bits based on Equation 1, toggling one and just one of these seven bits, can recover the consistency (i.e., each one of $2^3 = 8$ combinations of the check bits determines the state of no toggling or toggling just one of the seven bits). Hence, if just one of these bits is toggled, it can be found, and the original bit set can be recovered. In decoding phase, the Syndrome vector $S = [S_1 S_2 S_3]$ is generated from the received vector $[D_1, D_2, D_3, D_4, P_1, P_2, P_3]$ to determine the not satisfied parity-check equations in Equation 1. The Syndrome vector is calculated using Equation 2.

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

The Syndrome vector [0 0 0] means that no error is detected; otherwise, the single modified bit in the received vector is indicated by the Syndrome vector. For example, Syndrome vectors [1 0 1] or [0 1 0] address toggling D_2 or P_2, respectively. Based on the literature on information hiding (including steganography and watermarking), the low significant bits of the media (images here) are replaced by the check bits. As a weakness, toggling more than one bit cannot be detected and recovered by Hamming. For this reason, the watermarking methods based on Hamming code (7, 4) were not much successful in the literature except in [11]. The main idea of that work is distributing involved bits of a Hamming code in the image to be sure of having at most one toggling in the bit set after tampering with the watermarked image. Hence in certain conditions, the image can be perfectly recovered.

3 Related work

The first study, which is the base of many works for tamper detection and recovery, is briefly described here. Lin *et al.* [4] proposed a hierarchical scheme of fragile watermarking for tamper detection and recovery. In most of the related works of fragile watermarking, the image is first divided into non-overlapping blocks with specific sizes in pixels. In [4], the size of the blocks is 4×4 pixels, and each block contains four sub-blocks with a size of 2×2 pixels. In each pixel, the six Most Significant Bits (MS-Bits) play the role of data bits, and the two Least Significant Bits (LS-Bits)

are dedicated to replacing a watermark. The watermark of a sub-block A is used to detect modification and to recover data bits of another sub-block B. Hence, the four pixels in A have $4 \times 2 = 8$ bits to embed the watermark. Six of them are dedicated to the recovery bits, as the average of the intensity of the sub-block B considering just the six data bits. The other two bits authenticate the recovery bits (e.g., as parity check). In the tamper detection step, the authenticated watermark of each sub-block A is used to detect any modification in data bits of associated sub-block B. In the case that the average of data bits of four pixels in B is not the same as recovery bits in A, all four pixels are replaced by the averaged intensity. For more details, the reader is referred to [4]. By focusing on watermarking method based on watermark distributing, there are two major works [12] and [13]. Lee and Shunfeng [12] proposed a dual watermark scheme for detecting image tampers and possible recovery. Their proposed method is based on Lin's method [4]. In their method, the image is divided into upper and lower parts. The blocks have a size of 2×2 pixels, and each block in the upper half has a partner in the lower half. The watermark of each block is not only embedded in its own mapped block but also embedded in the partner block in order to provide two chances of recovery. It could achieve better results in recovery than Lin's method. In each 2×2 block, just 5 MS-Bits are used as data bits. Hence, the three least significant bits per pixel are allocated to embedding the watermark. The watermark length is, so, $4 \times 3 = 12$ bits. The average intensity of MS-Bits of each of paired blocks is stored as the watermark in these bits (one copy in each mapping block of paired blocks). Two remaining bits in each block are used for authenticating the stored watermark. Two schemes for image tamper detection and restoration were proposed by Sarkar *et al.* [13] in 2020. One of the schemes works in the frequency domain, and the other in the spatial domain. The quadruple watermarking approach has been implemented in a spatial domain scheme. In their proposed method, four chances are provided to recover the destroyed block. The watermark is generated from a set of four blocks $A_1, \ldots, A_4$ and is embedded in another set of four blocks $B_1, \ldots, B_4$, distributed in the image, using a mapping algorithm. The image is divided into 3×3 blocks of pixels with the six most significant data bits. The watermark length is, hence, $9 \times 2 = 18$ bits. The average intensity of each block A_1 (with a compressed form) generates the recovery bits. Finally, the watermark is embedded and duplicated in all sub-blocks B_1. Chan and Chang [14] and Chan [15] proposed an image authentication method using the Hamming code technique. In Chan and Chang's method [14], check bits are divided into two groups based on the most significant bits of each data bit so that in each group, the check bits are unique. As a result, if the value of the most significant bit per pixel is recognized, the total pixel value is obtained. Since the pixel value changes significantly with the wrong prediction of the most significant bit per pixel, Chan proposed his method by reducing the effect of this prediction. They first rearrange the data bits and then check bits are generated. It creates a new grouping of check bits such that the value of the most significant bits in each pixel was deterministically specified. However, due to that data bits are stored in the same pixel and also check bits are not separated, any pixel toggling may change more than one bit. Consequently, detection may be missed, and recovery may be incorrect. This is why; that work just was used for tamper detection. In other words, the special design and capability of Hamming code were not used. In this paper, The Hamming code is merged with a distributing method to form a powerful recovering method. Based on our knowledge, although there are methods that claim high recovery quality [16], but do not guarantee a perfect recovery of the watermarked image, except for the one proposed in [11]. In that method, not only is Hamming code used to generate the watermark, but also its property was applied in temper detection and recovery. In general, their method is based on distributing the bits involved in Hamming code. For this purpose, the image is first divided into eight independent parts. Then, all bits of the image are grouped in a set of distributed Hamming codes that contain one bit from each part of the image. In that method, Hamming code (7,4) was extended to Hamming code (8,5). It means that, by five data bits which are selected from separate parts of the image, three check bits of the watermark are generated and embedded in the three remained parts. Data bits are selected from the most significant bits, and the check bits, which form the watermark, are embedded in the least significant bits. Since in that method, each bit of the Hamming code is in a separate part and is distributed in the image, if the tamper was smaller than a quarter of the image in diameter, at most one bit of each Hamming code can be destroyed. According to the Hamming code property, the tamper is detected and perfectly recovered. That method has some problems in embedding. Tampering specific parts of the image may significantly degrade the recovery. Also, if the tamper is larger than the threshold, more than two bits may be toggled. Hamming code can correct up to one error bit. Hence, the corresponding Hamming code not only cannot detect the toggled bit but also destroys another bit in another part. In this paper, this method has been improved in order to tackle these problems. By considering the dependency between Hamming codes distributed in common pixels, a method is proposed to recover even two bits of code. Hence, a tamper smaller than half of the image in

width and height can also be recovered.

4 Proposed method

In this section, the novel fragile watermarking [11] and the proposed improvements, including three stages (embedding, detecting, and recovery), are presented in detail. In the embedding stage, it is explained how the watermark is generated using Hamming code, distributed, and embedded in some least significant bits of the image. In the temper detection and recovery stages, the Hamming code property is used to detect and reconstruct the temper in the image perfectly in specific conditions.

4.1 The embedding procedure

In grayscale images, the intensity of each pixel is presented by eight bits giving 256 possible different gray values. It has been addressed in Section 2 that; the four Most Significant Bits (MSBs) of data are encoded by Hamming (7,4) into three check bits embedded in three Least Significant Bits (LSBs). Hence, the fifth most significant bit is used neither as data nor a check bit. Having no information about the fifth MSB in the watermarked image degrades the quality of recovery. This is why; Hamming (8,5) was proposed in [11] and used in this paper, as presented in Equation 3.

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

In other words, XOR of D_4 and D_5 plays the role of the 4^{th} data bit in traditional Hamming code. If just one bit is toggled, it may be exactly or probably (with the chance of 50%) determined. With this modification, in the case of having all check bits inconsistent with the equations in Equation 3, called Full Inconsistency, one of D_4 or D_5 has been toggled. Otherwise, the toggled bit $(D_1, D_2, D_3, P_1, P_2, P_3)$ is deterministically known. As the next contribution, the Hamming code is applied on a Distributed Pixel (DP), which includes eight bits from eight different and sufficiently far pixels in the image. These pixels are called original pixels, in the rest of the paper, to be distinguished from DPs. A set of eight far original pixels in the image form a Distributed Block (DB). From each DB, eight DPs can be extracted. To generate watermarks based on the modified Hamming code (8,5) on a DP, five data bits are selected from five MSBs of different original pixels, and the check bits are embedded in three LSBs in three other pixels. Hence, each original pixel, five and three times, plays the role of data and check bit provider, respectively.

Figure 1 shows two sets of far pixels as two DBs. As mentioned, each DB produces eight different DPs.

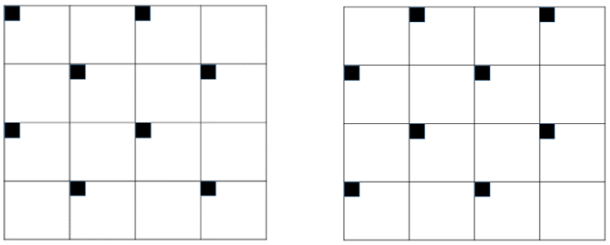

Figure 1. Two templates of selecting distributed pixels in the embedding procedure(7,4)

Hence the bits of each DP are also far from each other. These templates can rotate to produce other DBs. More precisely, as shown in Figure 1, the Euclidean distance of two original pixels of a DB is at least T equal to a quarter of the diameter of the image. For example, in a square 256×256 image, the length of the diameter of the image is $255\sqrt{2}$; as a consequence, $T = \dfrac{255}{2\sqrt{2}} \approx 90.16$. This threshold is the maximum size of the tamper in diameter to guarantee the perfect reconstruction in [11], where the diameter of a tamper is defined as the longest distance between two modified pixels in the image. With this design, if the diameter of the tamper is less than T, two different original pixels of a DB are never modified simultaneously. Because one and just one bit from each original pixel participates in a DP; at most, one bit of a DP can be modified. Hence, if the proposed Hamming code is applied on each DP, the modification can perfectly be recovered unless in the case of Fully Inconsistency. In addition, two horizontally/vertically neighbor pixels of a DP are at least as far as one-half of the width/height of the image. It means that a tamper smaller than half of the image in width and height can cover at most two orthogonally neighbor pixels. Also, if four horizontal or vertical parts of the image have been tampered, just two pixels have been destroyed. One of the main merits of the proposed method in this paper rather than [11] is recovering even two bits of error by considering dependencies of DPs in a DB. With this extension, the size of guaranteed perfect recovery is four times the one in [11]. Inspired by [11], from all possible choices, a simple approach was used in two rounds. In the first round, the four different original pixels from the upper half of the image (called the upper part) are considered to provide four data bits of associated DPs. Then, one pixel of the lower half of the image (called the lower part) is also selected as the provider of the last data bit. This procedure is repeated four times with different unused MSBs of associated pixels. Each pixel of the lower part plays the role of data bit provider just once. In each iteration, corresponding check bits are generated and overwritten on the associated LSBs of other unused three pixels from the lower part. In the second round, the roles of pixels in the upper and

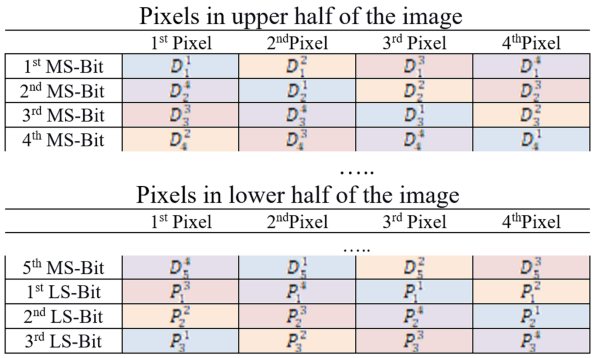

Figure 2. The first round of embedding: four MSBs and LSBs of pixels in the upper and the lower half of the image, respectively, in the embedding procedure(7,5)

lower parts are swapped.

This scheme is depicted in Figure 2 where D_k^i and P_k^i are k^{th} data and check bits, respectively, in Hamming code presented in Equation 3 and i represents the iteration of the first round. Note that the first and the third pixels correspond to the first and the second pixels of the first row, and the second and fourth pixels correspond to the first and second pixels of the second row in each part. In the second round, the same pattern is executed to embed MSBs of pixels in the lower part. The involved bits of the same DP (or iteration) are colored similarly. As shown, in the first round, four MSBs of the pixels in the upper half of the image and one MSB and three LSBs of the pixels in the lower half of the image contribute to the coding. All MSBs (five bits) of pixels are considered the data bits, and LSBs are replaced by the check bits. In [11], in the first iteration, all the bits $D_1 - D_4$ were the most significant bit of the original pixels in the upper part. In the second iteration, these bits were the 2^{nd} MSB of these pixels and so on. As shown, however, in Figure 1, the orthogonal selection is proposed. The proposed bit selection in this paper has three merits rather than [11] as follows:

- The k^{th} Hamming bit is also the k^{th} bit of the associated original pixel. Hence 4^{th} and 5^{th} bits of Hamming code, which may be changed without deterministic detection, are not selected from the three MSBs.
- Each pixel in the upper and lower parts provides, respectively, 4^{th} and 5^{th} Hamming bits just once. Hence, uncertainty is distributed between the pixels uniformly.
- The 4^{th} and 5^{th} bits in a DP are far with the distance of 2T. Although Hamming Code can recover any single-bit modification, in the extended Hamming Code (8,5), 4^{th} and 5^{th} bits can be simultaneously changed without making DP inconsistent. With the proposed bit selection, even in the case of having tamper sizes greater than T but less than 2T in diameter, 4^{th} and 5^{th} bits cannot be modified simultaneously.

These properties of the proposed embedding are some of the differences between this work with the one presented by the authors in [11]. However, the main contribution is to detect the modification of two bits of a DP in the recovery procedure.

4.2 The tamper detection procedure

In order to find the location of the tampered area, first, the watermark is extracted. Hence, the Syndrome vector is generated by Equation 2 For example, the Syndrome vector is computed in Equation 4 for the i^{th} iteration (DP) of each round.

$$\begin{aligned} S_1^i &= D_1^i \oplus D_2^i \oplus D_4^i \oplus D_5^i \oplus P_1^i \\ S_2^i &= D_1^i \oplus D_3^i \oplus D_4^i \oplus D_5^i \oplus P_2^i \\ S_3^i &= D_2^i \oplus D_3^i \oplus D_4^i \oplus D_5^i \oplus P_3^i \end{aligned} \tag{4}$$

Suppose the tamper is such that it modifies up to one bit in each DP. In this case, the Syndrome vector with a value of [000] indicates no error has occurred, and the Syndrome vector with a value of [111] indicates one of D_4 or D_5 has been modified. For the rest of the Syndrome vector values, one of the bits D_1, D_2, D_3, P_1, P_2 or P_3 has been modified. One of the most merits of the proposed method in comparison with [11] is its extension to reconstruct even two bits of error in each DP. For simplicity, this extension is described in Hamming code (7,4). Based on four data bits, sixteen (2^4) consistent 7-bit Hamming codes can be generated. Hence, there are totally $2^7 = 128$ codes out of which 12.5% are consistent. If it is assumed that, at most one error bit has occurred, there are eight possible observations for each consistent code after tampering (no error or one of the seven bits is toggled). Let us call this set of observations a candidate set. In Hamming method, candidate sets of two consistent codes have no common code. This is why; for each observed code, there is only one consistent code from which this observation can be generated. However, if there are two bits of error, Hamming code's assumption is wrong and certainly determines another bit for correction. It leads to achieving a code with three error bits. In the method in [11], tampering two far pixels contributing in a Hamming code leads to distortion of a pixel somewhere else after reconstruction. However, in this paper, it is assumed that, at most, two bits may be modified. Hence, there are 29 = possible observations in each candidate set. Of course, a pair of candidate sets now have common codes. Also, it can be shown that each code is seen in four candidate sets if itself is inconsistent. But, a consistent code cannot be seen in other candidate sets. Consequently, each

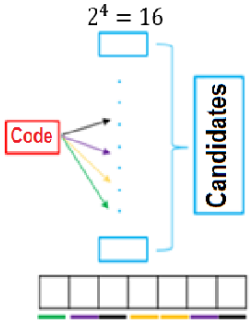

Figure 3. Four candidate sets which contain the inconsistent code, in the tamper detection procedure

inconsistent observed code may be generated from four consistent Hamming codes. One of these codes has one bit of difference and the other three codes have two bits of difference from the observed code. An example is shown in Figure 3.

Based on this approach, an inconsistent distributed Hamming code proposes four solutions, each of which addresses different bits (in different original pixels) for toggling. The challenge is now, which one of these solutions should be selected? This code is one of the eight DPs extracted from a common DB of the image. Each of these Hamming codes proposes its own solutions, and each solution determines one or two original pixels of the image as tampered pixels. It is assumed that at most, two out of the eight original pixels can tamper. Hence, the tampered pixels should cover a solution for each DP. These pixels are considered tampered pixels, and associated solutions are applied to reconstruct them.

4.3 The tamper recovery procedure

As mentioned, two-bit modifications can also be discovered if the tamper size is less than half of the image in width and height or less than a quarter of the image in width or height. Unlike the traditional Hamming code in [11], which corrects single-bit modifications, this paper attempts to provide a method to correct more bits of error in each DP. Assuming two out of eight bits are modified, twenty-eight (i.e., (82) = 28) different combinations of pixel distortion in a DB are possible. Each combination is called, here, a Tamper Combination (TC). Also, with five data bits, $2^5 = 32$ original consistent DPs, called Hamming String (HS), are possible. Given two sequences of bits, their Hamming Distance (HD) is defined as the number of corresponding bits that are different. Given eight bits of a DP, only four HS exist with HD not greater than two. Each candidate HS is associated with one (if the distance is two) or more (if the distance is one) specific TCs. Therefore, a limited set of TCs can be extracted for a DP. As explained, each DB is involved in forming eight different DPs. The true TC is computed by intersecting on candidate TCs of each DP in a DB. Finally, each DP is recovered by the selected TC and is replaced by the associated Hamming String.

5 Experimental results

The performance of the proposed scheme is measured, in this section, considering both tamper detection and image recovery in comparison with related state-of-the-art methods. For quantitative evaluation, Peak Signal-to-Noise Ratio (PSNR) is a well-known criterion in watermarking to evaluate the quality of image I_1 relative to image I_2 as defined in Equation 5 and Equation 6.

$$PSNR = 20 \times \log\left(\frac{255}{\sqrt{MSE}}\right) \tag{5}$$

$$MSE = \frac{1}{M \times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}|I_1(i,j) - I_2(i,j)|^2 \tag{6}$$

The symbols $I_1(i,j)$ and $I_2(i,j)$ addresses the pixel values in i^{th} row and j^{th} column of the image I_1 and I_2 respectively. The width and the height of the image are also denoted by M and N pixels, respectively. In the absence of noise or perfect recovery, the two images I_1 and I_2 are identical, and thus the MSE is zero. In this case, the PSNR is infinite, but due to prevent ambiguity, this case is reported as "Perfect Recovery" in the experiments. In addition, Structural Similarity Index Measure (SSIM) has been recently proposed to measure the similarity of an image with its original version based on the structure of the image, not bit errors as shown in Equation 7

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{7}$$

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 denominators and are set to default values presented in [17]. As can be seen, SSIM compares the total similarity of the images with respect to the mean, variance, and covariance of pixel values. It also has less concentration on specific tampered pixels in comparison with PSNR. This problem, along with having no SSIM for most of the 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 4, which are "Lena", "Cameraman", "Pepper" and "Barbara". All these gray-scale images are of size 512 × 512 pixels.

Figure 4. Some most widely used standard test images (Original Images): "Lena", "Cameraman", "Pepper" and "Barbara"

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

Original image	PSNR
pepper	39.10
Lena	39.09
Barbara	39.11
Cameraman	39.12
Plane	39.13
Baboon	39.08
Boat	39.07
Zelda	39.11
Elaine	39.09
Home	39.08

When the watermark is embedded in the original image, the watermarked image is in hand. It is very important that the watermarked image is not visually different from the original image, meaning that the original image, after embedding the watermark, should have acceptable quality. Figure 5 shows the watermarked image of the original image in Figure 4. As it is clear, the visual differences between the original image and the corresponding watermarked image are not discernible. As mentioned, embedding PSNR measures imperceptibility and similarity between the original and the watermarked images. Table 1 demonstrates the PSNR of watermarked images for ten standard images, including the ones presented in Figure 4 and Figure 5.

Significant results could be achieved by the proposed

Figure 5. Watermarked Images of Figure 4

(a) (b) (c)

Figure 6. Image of Barbara: (a) 3% tampered; (b) the detected tampered regions; (c) recovered image

(a) (b) (c)

Figure 7. Image of Pepper: (a) 6% tampered; (b) the detected tampered regions; (c) recovered image

method in tamper detection and recovery, particularly if the tempered size does not exceed the threshold T. Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, and Figure 11 show the result of the recovered image relative to various tampered sizes.

In Figure 11-(a), 25% of the pixels of the image are tampered. Figure 11-(b) and Figure 11-(c) indicate the tamper detection results using two templates of DBs presented in Figure 1. Finally, the perfectly recovered image is presented in Figure 11-(d). In order to further compare the performance of the proposed scheme with the related works, the PSNR of the recovered image vs. the watermarked image is reported in Table 2 and Table 3 with various tamper sizes. It measures

(a) (b) (c)

Figure 8. Image of Cameraman: (a) 10% tampered ; (b) the detected tampered regions; (c) recovered image

(a) (b) (c)

Figure 9. Image of Lena: (a) 15% tampered; (b) the detected tampered regions; (c) recovered image

(a) (b) (c)

Figure 10. Image of Barbara: (a) 20% tampered; (b) the detected tampered regions; (c) recovered image

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

Figure 11. The "Lena" image (a) with 25% tampered pixels, (b) and (c) the detected tamper in accordance with DB patterns in Figure 1, and (d) the recovered image

the ability of the methods in order to reconstruct the watermarked (not original, which is not in hand) image.

Table 2. The PSNR of the recovered image respect to the tampered size

Paper	Embedding rate	PSNR(dB) of Recovered image
Ref [15] method #1	3	51.97
Ref [15] method #2	3.25	52.90
Ref [15] method #3	3.49	53.42
Ref [15] method #4	4	Perfect Recovery
Proposed method	3	Perfect Recovery

Table 3. The PSNR of the recovered image respect to the tampered size

Paper	10%	20%	25%
Ref [12]	35.17	–	33.45
Ref [13] (Quad)	41.10	39.45	–
Ref [13] (DWT)	45.34	41.23	–
Ref [18]	45.85	–	–
Ref [19]	37.50	–	33.95
Ref [5]	38.69	37.15	–
Ref [20]	45.09	40.58	39.50
Ref [21]	44.15	41.83	–
Ref [22]	48.21	45.07	–
Ref [11]	∞	39.44	37.16
RRef [23]	40.48	36.57	34.80
Ref [24]	47.00	43.00	41.00
Ref [25]	36.00	30.00	29.00
Proposed method	Perfect Recovery	Perfect Recovery	Perfect Recovery

The PSNR of the recovery image is shown in Table 2 to compare the proposed method with four classic watermarking methods based on Hamming code [15]. Based on this reference, the tamper size is $64 * 64$ in $512 * 512$ images. The average number of bits used to embed the check bits as the watermark is called the embedding rate. The greater the embedding rate, the lower PSNR in the embedding stage. As shown, the method #4 can perfectly recover the image, but with a low PSNR of embedding. However, the proposed method along with maintaining the PSNR value of embedding the watermark in just three LS-Bits, achieves the perfect recovery.

The results of the proposed method, compared to some other related state-of-the-art methods, are reported in Table 3. The results indicate that the proposed method could outperform others. As shown, the proposed method is the only one that perfectly recovers the image for 10% tamper size. Although all the methods have not reported the PSNR for all tamper

Table 4. The SSIM of the recovered image respect to the tampered size

Original image	SSIM(up to 25%)
Pepper	100
Lena	100
Barbara	100
Cameraman	100
Plane	100
Baboon	100
Boat	100
Zelda	100
Elaine	100
Home	100

sizes, and the proposed method is originally created for perfectly small tampers, larger tampers are also investigated. The proposed method completely recovers the temper with a maximum size of 25% of the total image. SSIM has been reported in specific conditions in a few researches. Table 4 demonstrates the SSIM of recovery images with up to 25% tampered size.

6 Conclusion

In this paper, a new fragile watermarking method to detect the tampered area in the image and recover the original image has been proposed. Using Hamming code applied on distributed Pixels is the main contribution of this work. A Hamming (8,5) is proposed to generate the check bits. Then, it is integrated with a pixel-distributing scheme. The contribution in embedding is degrading the sensitivity of the image recovery to tampering with specific parts of the image. Also, the proposed reasoning in the recovery phase leads to reconstructing even two-bit modifications in Hamming with a high probability. Experimental results accepted this claim, and the proposed method completely recovered the temper with a maximum size of 25% of the total image. Focusing on decreasing the embedding PSNR, inserting the key to preserve embedding privacy, decreasing the embedding rate, and other bit or pixel selection schemes in distributed blocks can be investigated 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] 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.

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

[5] 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.

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

[7] Dipabali Sarkar, Sarbani Palit, Sukalyan Som, and KN Dey. Large scale image tamper detection and restoration. *Multimedia Tools and Applications*, 79(25):17761–17791, 2020.

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

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Faeze Rasouli and Mohammad Taheri. A new fragile watermarking based on distributed hamming code. In *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, pages 1–5. IEEE, 2021.

[17] 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.

[18] 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.

[19] 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.

[20] Omer Hemida and Hongjie He. A self-recovery watermarking scheme based on block truncation coding and quantum chaos map. *Multimedia Tools and Applications*, 79(25):18695–18725, 2020.

[21] Ziyun Xia, Wenyin Zhang, Huichuan Duan, Jiuru Wang, and Xiuyuan Wei. Fragile watermarking scheme in spatial domain based on prime number distribution theory. *Multimedia Tools and Applications*, 81(5):6477–6496, 2022.

[22] Afrig Aminuddin and Ferda Ernawan. Ausr1: Authentication and self-recovery using a new image inpainting technique with lsb shifting in fragile image watermarking. *Journal of King Saud University-Computer and Information Sciences*, 2022.

[23] Li Huang, Da Kuang, Cheng-long Li, Yu-jian Zhuang, Shao-hua Duan, and Xiao-yi Zhou. A self embedding secure fragile watermarking scheme with high quality recovery. *Journal of Visual Communication and Image Representation*, 83:103437, 2022.

[24] Payal Garg and Ajit Kumar Jain. Digital watermarking techniques and their analysis. In *Smart Systems: Innovations in Computing*, pages 41–54. Springer, 2022.

[25] Saeed Sarreshtedari, Aliazam Abbasfar, and Mohammad Ali Akhaee. A joint source–channel coding approach to digital image self-recovery. *Signal, Image and Video Processing*, 11(7):1371–1378, 2017.

Faeze Rasouli received her B.Sc. degree in hardware engineering from Hamedan University of Technology, Hamedan, Iran, in 2017. Now she is an M.Sc. student in the Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran. Her research interests are Database Security, Image Processing, Computer Vision, and 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) from 2001 until 2013. He started his job as a faculty member (2014) in that department with research in machine learning, fuzzy systems, large margin classifiers, optimization, modeling, and information hiding.

Reza Rohani Sarvestani was born in 1985. He received a B.Sc. in computer science from Bahonar University, Iran, in 2006 and an M.Sc. and Ph.D. in artificial intelligence from Shiraz University, Iran, in 2009 and 2016, respectively. He started his job, as a faculty member (2016) at Shahrekord University, Iran, with research in machine learning, Data fusion, biomedical signal processing, and speech processing.