ISSN: 0067-2904

Image Watermarking Based on IWT and Parity Bit Checking

Jalal H. Awad[1*], Balsam D. Majeed[2]
[1]Department of Computer, College of Science, Mustansiriyah University , Baghdad, Iraq
[2]Department of Computer techniques Engineering, Imam Kadhim Faculty of Islamic Sciences University, Baghdad, Iraq

Received: 3/7/2020 Accepted: 14/10/2020

Abstract

Various document types play an influential role in a lot of our lives activities today; hence preserving their integrity is an important matter. Such documents have various forms, including texts, videos, sounds, and images. The latter types' authentication will be our concern here in this paper. Images can be handled spatially by doing the proper modification directly on their pixel values or spectrally through conducting some adjustments to some of the addressed coefficients. Due to spectral (frequency) domain flexibility in handling data, the domain coefficients are utilized for the watermark embedding purpose. The integer wavelet transform (IWT), which is a wavelet transform based on the lifting scheme, is adopted in this paper in order to provide a direct way for converting image pixels' integer values to integer coefficient values rather than floating point coefficients that could be produced by the traditional wavelet transform. This direct relation can enhance the processed image quality due to avoiding the rounding operations on the floating point coefficients. The well-known parity bit approach is also utilized in this paper as an authentication mechanism, where 3 secret parity bits are used for each block in an image which is divided into non-overlapped blocks in order to enforce a form of fragile watermark approach. Thus, any alteration in the block pixels could cause the adopted (even) parity to be violated. The fragile watermarking is achieved through the modification of least significant bits ((LSBs) of certain frequency coefficients' according to the even parity condition. In spite of this image watermarking operation, the proposed method is efficient. In order to prove the efficiency of our proposed method, it was tested against standard images using measurements like peak signal to noise ratio (PSNR) and structural similarity index (SSIM). Experiments showed promising results; the method preserves high image quality (PSNR≈ 44.4367dB, SSIM≈ 0.9956) and good tamper detection capability.

Keywords: video, watermarking, parity bit, IWT.

تعليم الصورة بالعلامة الما ئية بناءً على تحويل الموبج الصحيح وفحص بت التكافؤ

جلال حميد عواد *[1]، بلسم ضياء مجيد [2]
[1]قسم الحاسوب، كلية العلوم، الجامعة المستنصرية، بغداد، العراق
[2]قسم هندسة تقنيات الحاسوب، كلية الإمام الكاظم للعلوم الإسلامية الجامعه، بغداد، العراق

*Email: jalalhameed@uomustansiriyah.edu.iq

الخلاصة

تلعب أنواع المستندات المختلفة دورًا مؤثرًا في الكثير من أنشطة حياتنا اليوم؛ ومن ثم فإن الحفاظ على سلامتهم أمر مهم. وتنوعت هذه الوثائق من نصوص ومقاطع فيديو وأصوات وصور. ستكون مصادقة الأنواع الأخيرة مصدر اهتمامنا هنا في هذه الورقة. يمكن معالجة الصور مكانيًا عن طريق إجراء التعديل المناسب مباشرةً على قيم البكسل الخاصة بها أو طيفيًا من خلال إجراء بعض التعديلات على بعض المعاملات التي تمت معالجتها. نظرًا لمرونة المجال الطيفي (التردد) في معالجة البيانات، يتم استخدام معاملات المجال لغرض تضمين العلامة المائية. تم اعتماد التحويل المويجي الصحيح (IWT) وهو تحويل مويجي قائم على مخطط الرفع في هذه الورقة من أجل توفير طريقة مباشرة لتحويل قيم الأعداد الصحيحة لوحدات بكسل الصورة إلى قيم معامل عدد صحيح بدلاً من معاملات النقطة العائمة التي يمكن إنتاجها بواسطة التحويل المويج التقليدي. هذه العلاقة المباشرة يمكن أن تعزز جودة الصورة المعالجة بسبب تجنب عمليات التقريب على معاملات الفاصلة العائمة. يتم أيضًا استخدام نهج بت التكافؤ المعروف في هذه الورقة كآلية مصادقة ، حيث يتم استخدام 3 بتات تكافؤ سرية لكل كتلة في صورة مقسمة إلى كتل غير متداخلة من أجل فرض شكل من أشكال نهج العلامة المائية الهشة. وبالتالي ، فإن أي تغيير في وحدات البكسل يمكن أن يؤدي إلى انتهاك التكافؤ (الزوجي) المعتمد. تتم عملية وضع العلامات المائية الهشة من خلال تعديل LSBs لمعاملات التردد وفقًا لشرط التكافؤ المتساوي. على الرغم من عملية وضع العلامات المائية على الصورة ، فإن الطريقة المقترحة فعالة. لإثبات كفاءة طريقتنا المقترحة ، تم اختبارها مقابل الصور القياسية باستخدام قياسات مثل نسبة الإشارة إلى الضوضاء (PSNR) ومؤشر التشابه الهيكلي (SSIM). أظهرت التجارب نتائج واعدة. الطريقة المقترحة تحافظ على جودة الصورة العالية (PSNR≈ 44.4367dB، SSIM≈ 0.9956) وقدرة جيدة على اكتشاف العبث.

1. Introduction

As a consequence of the various imaging technology developments, new and advanced instruments have emerged. Such instruments may be a double-edged sword; hence, outlaws can utilize them for their illegal aims [1]. A tiny malicious change in a formal document copy may lead to a lot of catastrophic effects [2]. Therefore, as a countermeasure for such deceives, image tamper detection and tamper localization techniques were invented [3]. An old-fashioned method to ensure image integrity is to utilize the digital signature, but unfortunately this cannot help in determining the tampered regions if any exist [2, 4]. Therefore the art of using image watermarking imposes itself here to act as authentication and, perhaps, a tamper location identification tool. The watermark technique should preserve as much as possible the watermarked image quality such that the outlaws will not observe the watermark presence [5].

Watermarks can be categorized into three broad categories; robust, fragile, and semi-fragile. Robust watermarks can serve for copyright protection because of their robustness (resistance) against intentional or unintentional modification of the watermarked image; fragile watermarks are susceptible to destruction in case of intentional or unintentional watermarked image modification [2, 6], the semi-fragile category is intended to resist only unintentional modifications like JPEG compression [7].

In fragile watermarks, tampering causes the destruction of the hidden watermarks of each tampered region in the cover image (this type is used in this research to watermark the images). This in turn alerts the receiver about the existence of tampered regions, as the comparison of the extracted corrupted watermark indicates frame region(s) alteration [8].

Watermarking algorithms can utilize spatial domains in order to conceal watermark bits, usually through substituting them with original pixels' LSBs [1, 9]. Other algorithms utilize the principle frequency domain coefficients to hide their watermark bits [6], thus gaining the opportunity of only dealing with the most important coefficients and hence gain the reduction of processing time. The use of frequency domain leads to compact representation of the image visual information [1, 9].

The spectral domain (wavelet transform) is also concerned in this research.

2. Wavelet transform

Watermark embedding approaches can be divided into two categories; One category is the usage of spatial domain techniques, where the watermark bits are embedded directly into the pixels, which may

lead, from one hand, to degrade cover image quality and hence the risk of intruder perceptibility and, from the other hand, to less host (cover) robustness against unintentional changes, like jpg compression. The other category is the usage of frequency domain techniques like discrete Fourier transform (DFT), discrete cosine transform (DCT), wavelet transforms …etc. Some of these transforms, for example the DFT, have limitations, since they can give only frequency without any time information, which will restrict us to deal only with stationary (frequency doesn't change in time) signals. Images of our concern here are considered as non-stationary signals; therefore, an alternative frequency domain transform that can give both frequency and time information should be considered, and wavelet is such a type [10]. Wavelet transform depends on two filter types: low and high pass filters. As a consequence, the application of these two filter types gives low and high frequency information which represent the first level decomposition. The most important image information exists in the low frequencies, while edges and noise can be isolated inside some of the high frequencies. The same used filters can be used repeatedly on the resulted low information in order to reach another level of decomposition. At each level, a band of frequencies can be decomposed into approximation (LL) sub band, which holds the low frequency information, and three details (HL,LH and HH) sub bands, which hold the moderate and high frequency information. According to this, we can achieve and deal with the required frequency with a high level of precision [4]. Figure-1 below shows the first level (a) and second level (b) decompositions.

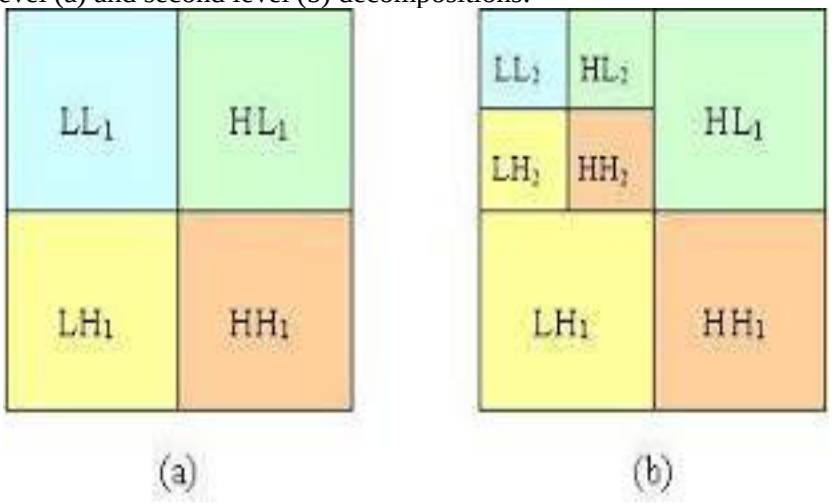


Figure 1- (a) first level wavelet transform, where an image is decomposed into four sub bands which are approximation (LL_1), vertical (HL_1), horizontal (LH_1) and diagonal (HH_1). (b) second level wavelet transform decomposition, where LL_1 is decomposed further into (LL_2, HL_2, LH_2 and HH_2).
Figure-2 below shows a 256 grayscale image decomposed into the second level.

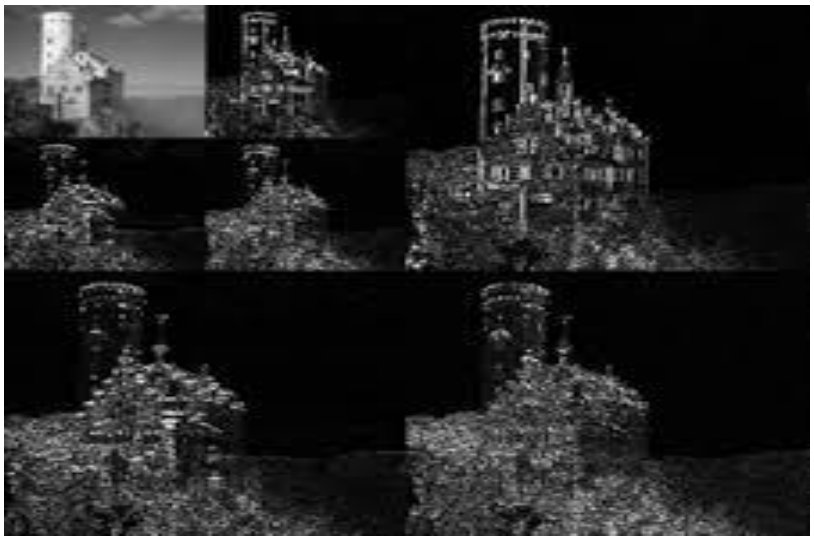


Figure 2- 2nd level wavelet transform. The upper left corner image is the approximation of the original image for the second wavelet transform level. This sub band has its complement sub bands in this level which are the three of the same size sub bands attached to it. The other bigger-sized sub bands are of the previous level; all are edges and have some noise, except the approximation which contains the most important information.

As can be seen from Figure-1, the approximation (LL) sub band holds the more crucial image information than the other sub bands. Also, since compression techniques need to preserve such information untouched and tend to handle the others to get compact representation, hence, it is motivating to conceal the watermark information inside this approximation sub band. Exploiting such privilege also has its side effects by losing some watermarked image quality, due to the exhaustive replacement of some of the most important (approximation) information with the watermark bits, which may result in less imperceptibility. Therefore, it may be necessary to adventure (we may lose these watermark bits) by embedding some watermark bits inside the other sub bands, because the compression algorithms in their nature tend to handle these bands. As a result of these compression operations, there will actually be some loss in the data and certainly, among them, some watermark bits. As a consequence, the watermarked image can have high imperceptibility (because Human Visual System can't notice the changes that take place on high frequency bands while easily pick up such changes in the low frequency bands), as well as achieving a high robustness against unintentional change through compression [10]. Discrete wavelet transform (DWT) low-level sub bands can be used in order to embed watermark bits [9]. As the DWT coefficients have real values in contrast to the image pixels which have integer values, this may require some quantization in these coefficients in order to be compatible (integer) with the image pixel values to have more flexibility in swapping between them through the embedding process. Such quantization will of course cause some information loss, therefore an alternative transformation approach, such as the Integer Wavelet Transform (IWT), will be suitable here [3]. Thus in this article, the IWD is an adequate choice for the embedding purpose. The embedded watermark bits, i.e. "0" or "1" in this research, will be calculated using a parity check bit generation method.

3. Parity bit

Authentication data can be generated using the parity check approach [11, 12]. Parity bit checking is a straightforward watermarking approach [13]. According to the parity type (even or odd), in order to embed the watermark inside the host, it is necessary to modify the host spatial or frequency coefficient values to have the suitable type according to a preceding deal between the transmission sides [14]. Parity (even/odd) checking is an approach that is used in networks to ensure the transmitted message correctness at the receiver side. The two transmission sides should beforehand agree on the parity type (even or odd). For each chunk of the transmitted data, the sender system must represent the data in binary form, then count the number of the ones and adds zero or one to the end of the chunk as a parity bit, making the number of 1s toward the agreed parity type. At the receiver side, the system re-computes the number of 1s to see if it satisfies the agreed parity; if not, the system declares incorrect data. In this paper, we will use the same parity bit manner to randomly generate 3 bits for each block as a key, such that these key bits enforce an even parity on the sent data by altering the LSB if the sent data doesn't satisfy the even parity. They also preserve the LSB as it is, if the data satisfies the even parity. This occurs at the embedding phase at the sender side. At the receiver side, the receiver system uses the key bits to test the data parity, accept those with even parity as non-tampered, and indicate the others that do not satisfy this even parity as tampered. The even parity (even number of "1s") will be used here.

The proposed embedding algorithm is as follows:

1- Read an image G.

2- Divide G into 8x8 pixels non overlapped blocks.

3- Make first level wavelet transform on each individual block to extract approximation (LL1), horizontal (LH1), vertical (HL1), and diagonal (HH1) bands.

4- Make another round of wavelet transform on each band (excluding HH1 band) in order to extract their sub bands (2nd level wavelet transform).

5- For each sub band (LL2, LL_{HL1} and LL_{LH1}, which are the low-low of LL1, HL1, and LH1, respectively), use the greatest coefficient to be transformed into binary representation B.

6- Concatenate each parity bit W (which is a randomly generated binary bit saved as a key) to B and test if the resulted series has an even parity of 1s. If not, change the LSB of B to its contrary.
7- Repeat steps 3-6 for each block.
8- Make the inverse integer wavelet transform (IIWT); by utilizing these adjusted sub bands (LL2, LL_{HL1} and LL_{LH1}) with their complement sub bands in order to form the higher level bands (LL1, HL1 and LH1). Then, make another IIWT on these latter bands in order to have the final watermarked image.
Figure-3 below shows the embedding phase.

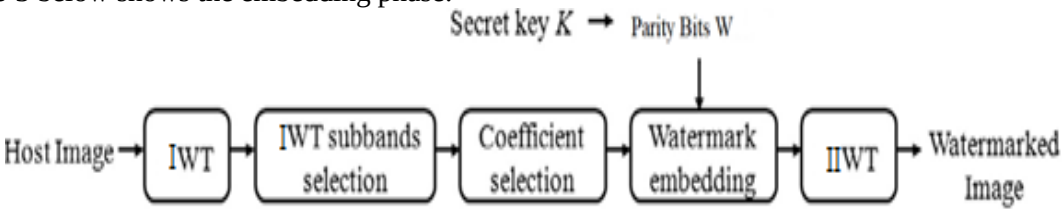


Figure 3- The watermark embedding phase, where a series of secret parity bits is generated to be used to moderate the greater coefficients in each 8*8 non-overlapped block.

The step-by-step details of the above algorithm are as follows:
Step 1: an image to be watermarked is read and converted to grayscale in case of the colored images, since this algorithm deals with images with one layer. Therefore, the colored image which naturally has three channels (R, G, B) has to be converted to a grayscale image which has just one layer.
Step 2: in order to have good local control on the overall image regions (blocks) and provide the appropriate immunity for them against tampering, it is necessary to divide the image into tiny regions (blocks) of 8 by 8 pixels. Thus, we can detect any tampering operation even in such small regions. For example, if we have an image of 512*512 pixels, then we can get 4,096 regions of 8*8 pixels, which is sufficient to detect any small change in any of these 4,096 regions of this image.
Step 3: for each of this 8*8 pixel regions, applying the wavelet transform will give four bands of 4*4 coefficients each. Thus, a first level of transformation from a spatial to a frequency domain occurs here. The reason for transforming this new domain is to have high degree of imperceptibility when hiding the watermark bits in such domain in comparison with the spatial domain. In nature, the Human Visual System is very sensitive to changes in spatial regions, especially in the flat (uniform) regions which, according to the frequency domain terminology, are called the low frequency regions. A solution for this spatial domain vulnerability can be done through the use of the frequency domain which has the ability to analyze each block into multi frequency bands (low-low, low-high, high-low and high-high). This will thus permit selecting and dealing with the appropriate band in order to hide watermark bits with a high degree of imperceptibility by any other un authorized third party. Figure-4 bellow shows the first level wavelet transform for a block into sub bands, each of 4*4 coefficients.

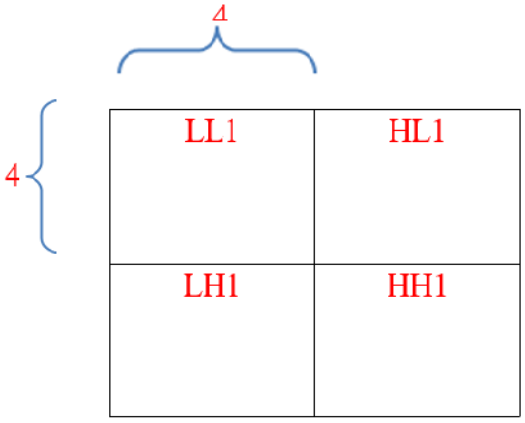


Figure 4- The first level decomposition of the 8*8 block into sub bands of 4*4, each using the wavelet transform.

Step 4: in order to have more flexibility in getting much more frequency levels and details, a second wavelet level is necessary here so as to getting a balance between imperceptibility and robustness requirements. A contradiction does exist between these two requirements, so we have to make a tradeoff between them. As we have stated before, the imperceptibility is related to the HVS and this, in turn, is highly sensitive to low frequencies, hence embedding our watermark bits inside these frequencies will degrade the imperceptibility. Avoiding this by hiding our watermark bits inside the high frequencies will solve this imperceptibility problem, but unfortunately will cause robustness degradation at the same time. This occurs due to the fact that compression algorithms tend to throw away a lot of the high frequency components through the compression operation, considering them as less important components, especially that a lot of noises are of high frequency nature. Hence, taking the low-low of the 2nd wavelet low and mid-level frequencies (LL2, LL_{HL1} and LL_{LH1}) will satisfy the proper selection of the appropriate frequencies for the embedding process, without defecting the imperceptibility or loosing important data through the compression operations. Figure-5 below shows the 2nd level wavelet transform sub bands of 2*2 coefficients for each sub band, with gray shaded sub bands that will be used for the embedding purpose.

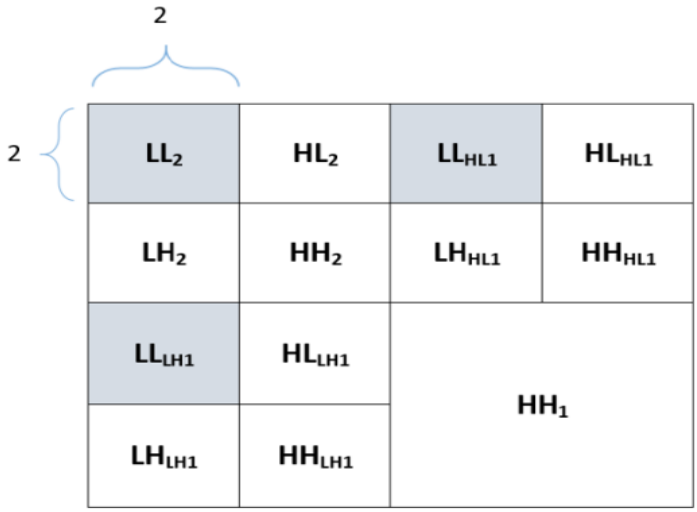


Figure 5- A second level decomposition of Figure-4 into sub band of 2*2 coefficients, indicating with shades the sub bands of our interest.

Step 5: the greatest coefficients in each sub band represent the other face of the coin to the most important edges (high frequencies) in the cover image. Also, because that the HVS is insensitive to changes in these regions, they represent a good choice for the embedding of the watermark bits.
Figure-6 below shows the selection of the greatest coefficients (blue shaded) and their binary representation for the purpose of the watermark bits embedding.

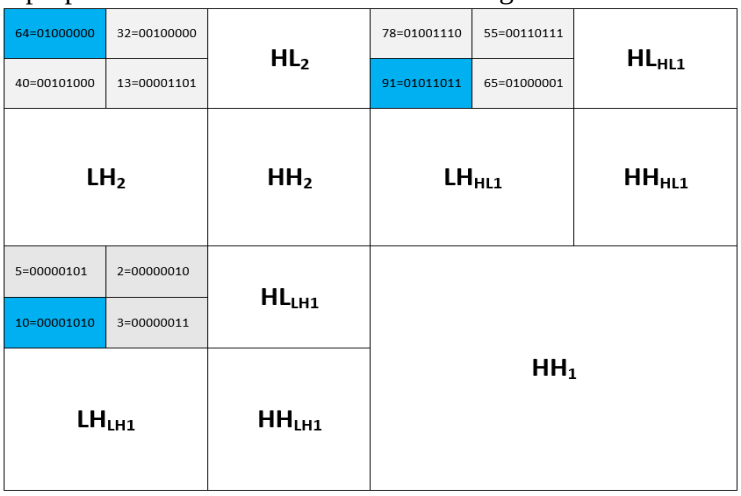


Figure 6-The greatest coefficients indicated with blue, in each of the selected sub bands of the second level wavelet decomposition. These coefficients are used for the purpose of watermark bits embedding.

Step 6: for each block, three randomly generated bits are saved and later used in the authentication process as watermark bits. Each of these 3 bits will be concatenated to the binary representation of their corresponding coefficients. Then, by checking the even parity of this binary string (including the watermark bit), it will be decided if it is necessary to change the LSB of the corresponding coefficient (from 0-to-1 or from 1-to-0) when the even parity is not satisfied after the concatenation. Otherwise, the LSB is left untouched and hence, in either case, the watermark bits are inserted. These 3 bits will be concatenated with their corresponding coefficients and it will be checked if the even parity is unsatisfied (tampered block) or satisfied (not tampered block) in the authentication phase.
Table-1 below shows numerical examples for the process of the image watermarking through enforcing the even parity on the combination of the coefficients' binary representation and the watermark bit.

Table 1-Tuning the coefficients' LSB by utilizing an extra watermark bit in order to enforcing the even parity on their combination. Note that for the coefficient with the decimal value of 64, there is no need to change its LSB because it achieves the even parity when combined with the parity bit. This is not the case for both the coefficients of 91 and 10 values.

Coefficient decimal value	Coefficient binary representation	Randomly generated parity bit (watermark bit)	Concatenated string (including watermark bit)	Needs LSB change-e	Watermarked coefficient	Watermarked coefficient decimal value
64	01000000	1	010000001	no	01000000	64
91	01011011	0	010110110	yes	01011010	***90***
10	00001010	1	000010101	yes	00001011	***11***

While Figure-7 shows the idea of implementing the checking of the even parity for the embedding phase.

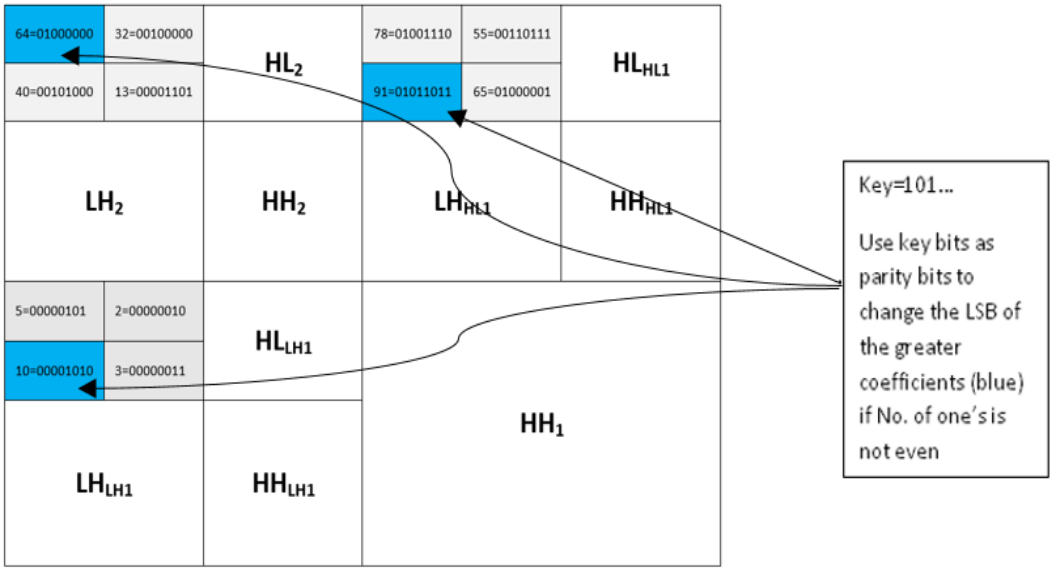


Figure 7-Even parity investigation of the greatest coefficient (blue) before changing LSB, if required, to be qualified for even parity.

Step 7: the previous steps (3-to-6) are required to be performed for each of the 4,096 blocks.
Step 8: after finishing the watermark bits embedding for all their corresponding coefficients in each block, it comes the turn to get the final watermarked image. This is achieved by first applying the

IIWT on the second level of that we have obtained from IWT in step 4, using the second level sub bands, as well as the other IIWT on the resulted sub bands together with HH1, in order to obtain the final watermarked image, as shown in Figure-8 below.

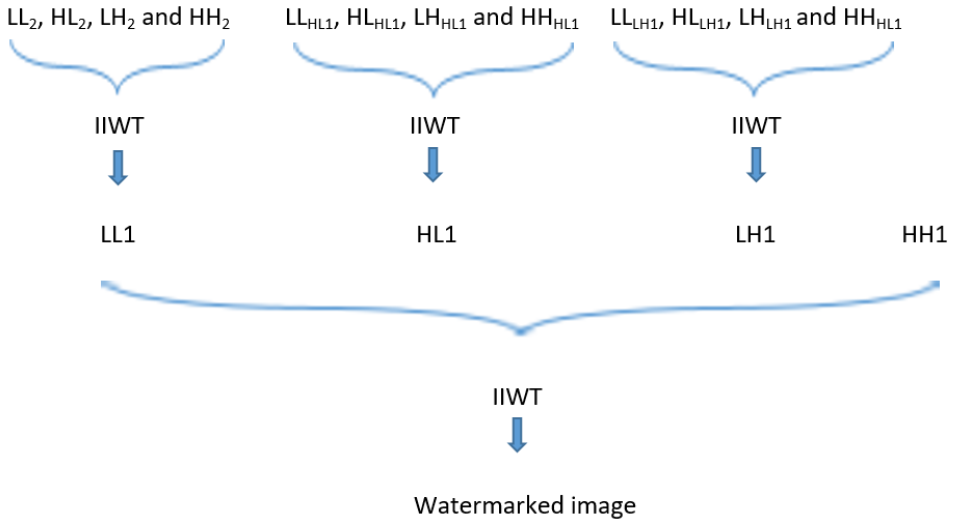


Figure 8- The backward wavelet transform after conducting an appropriate adjustment on the second level sub band coefficients.

The pseudo code for the embedding operation is given in Figure-9.

Algorithm1: the proposed embedding algorithm

1: **procedure** Embedding (***W, Cover image***).
2: ***Block*** ← divide ***Cover image*** into 8*8 non-overlapping blocks.
3: ***TBlock*** ← IWT (***Block***).
4: ***L2Block*** ← IWT (LL (***TBlock***)).
5: ***LL2*** ← LL (***L2Block***).
6: L_{HL1} ← IWT (HL (***TBlock***)).
7: LL_{HL1} ← LL (L_{HL1}).
8: L_{LH1} ← IWT (LH (***TBlock***)).
9: LL_{LH1} ← LL (L_{LH1}).
10: ***MLL2*** ← max (***LL2***).
11: MLL_{HL1} ← max (LL_{HL1}).
12: MLL_{LH1} ← max (LL_{LH1}).
13: ***DMLL2*** ← decimal2binary (***MLL2***).
14: $DMLL_{HL1}$ ← decimal2binary (MLL_{HL1}).
15: $DMLL_{LH1}$ ← decimal2binary (MLL_{LH1}).
16: ***CDMLL2*** ← concatenate (***DMLL2***, W_i).
17: $CDMLL_{HL1}$ ← concatenate ($DMLL_{HL1}$, W_{i+1}).
18: $CDMLL_{LH1}$ ← concatenate ($DMLL_{LH1}$, W_{i+2}).
19: if not even (***CDMLL2***) → if LSB (***DMLL2***) ==0 then LSB (***DMLL2***) ←1
Else LSB (***DMLL2***) ←0
20: repeat step ***19*** for $CDMLL_{HL1}$ and $CDMLL_{LH1}$
21: increment ***i***
22: ***LL1*** ← IIWT (LL_2, HL_2, LH_2 and HH_2).
23: ***HL1*** ← IIWT (LL_{HL1}, HL_{HL1}, LH_{HL1} and HH_{HL1}).
24: ***LH1*** ← IIWT (LL_{LH1}, HL_{LH1}, LH_{LH1} and HH_{HL1}).
25: ***watermarked Block*** ← IIWT (***LL1***, ***HL1, LH1***, HH1).
26: repeat steps ***3-25*** for each ***Block***.
27: ***watermarked image*** ← recombining all (***watermarked Block***).
28: End.

Figure 9- The pseudo code for the watermark embedding.

The proposed authentication algorithm includes the following:

1- The steps ***1-5,*** used in the embedding algorithm, are used here.

2- Each parity bit W is concatenated to B and we test if the resulted series has an even parity of 1s. If not, then the block is announced as tampered. Otherwise, the block is announced as not tampered.

3- The steps are repeated for each block.
4- The IIWT is made for two levels in order to get the final authentication resulted image, with its tampered blocks marked with black color.
Figure-10 shows the overall authentication phase diagram.

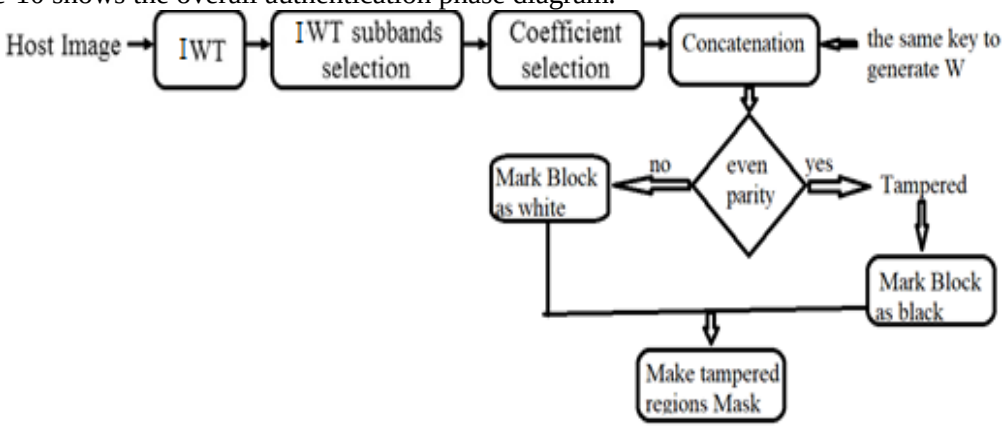


Figure 10-The extraction (authentication) phase.

The pseudo code for the authentication algorithm is given in Figure-11 below.

Algorithm2: the proposed authentication algorithm

1: **procedure** authentication (***W, watermarked Block***).
2: ***Block*** ← divide ***watermarked Block*** into 8*8 non-overlapping blocks.
3: the same steps from ***3-18*** of the embedding algorithm are used again
4: if not even (***CDMLL2*** or $CDMLL_{HL1}$ or $CDMLL_{LH1}$) then mark the ***Block*** as tampered by converting it to black color.
5: increment ***i***
6: repeat steps ***3-5*** for each ***Block***.
7: ***final authentication resulted image*** ← recombining all (***Block***).
8: End.

Figure 11- The authentication algorithm.

Figure-12 below shows the extraction phase, indicating tampered (red) coefficient due to tampering. Frequency domain coefficient are changed due to tampering in image spatial domain.

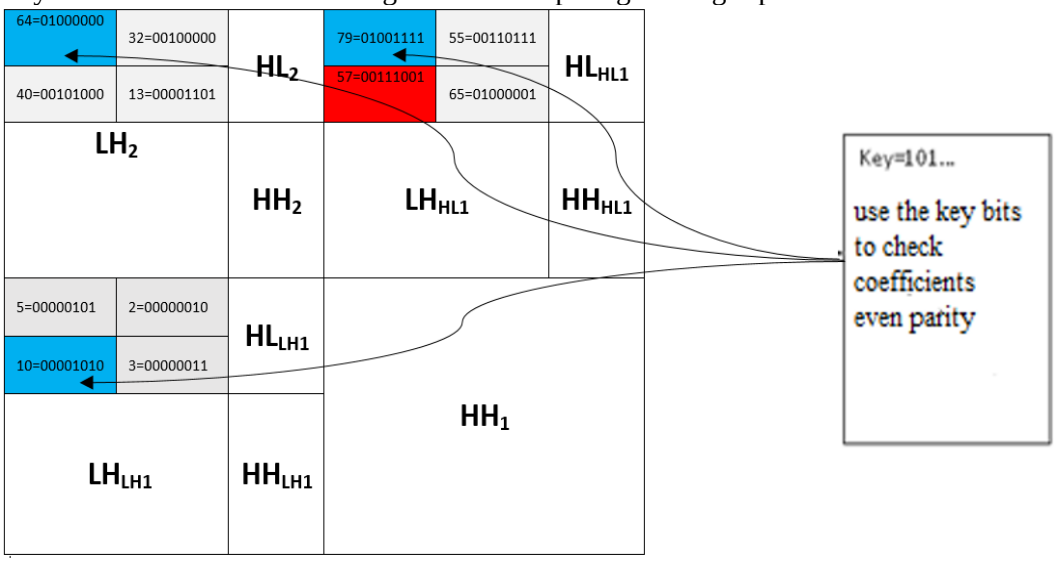


Figure 12-Tampering causes image region changes, hence some coefficients (red) change and therefore the greater coefficients and their parity may change. This block is the same block of Figure-

7), but it is a tampered version and hence some of its second level coefficients are changed (it was 91 but now is 57), which cause a selection of another coefficient and perhaps a violation of the even parity.

Figure-13 shows two versions of the same block coefficient values and their even parity checks before (a) and after (b) block tampering.

The greater coefficient decimal value	Coefficient binary representation	The block secret parity bit (watermark bit)	Concatenated string (including watermark bit)	Even parity violation	Tampered block
64	01000000	1	010000001	no	NO
90	01011010	0	010110100	no	
11	00001011	1	000010111	no	

(a)

The greater coefficient decimal value	Coefficient binary representation	The block secret parity bit (watermark bit)	Concatenated string (including watermark bit)	Even parity violation	Tampered block
64	01000000	1	010000001	no	***yes***
79	01001111	0	010011110	***yes***	
11	00001011	1	000010111	no	

(b)

Figure 13-(a) the 3 coefficients of non-tampered block and (b) tampering causes the alteration of at least one (as in this example) of the greater coefficients, also failing in the even parity constraint and therefore announcement of a tampered block.

4. Results

As image watermarking techniques depend on the embedding of some tiny information within the cover image, the latter will have some degradation in its quality. Therefore, innovated techniques tend to decrease such effect by implementing new approaches which try to combine the ability of having invulnerable, as well as imperceptible, watermark in this host. To measure imperceptibility, it is possible sometimes to consult experts to give their impression (subjectively). Unfortunately, such approach is expensive and may be imprecise. Therefore, the metrics of objective PSNR and SSIM (Structural Similarity Index) of image quality are used.

PSNR can be computed for two images (G1, G2) of equal nxm size and 255 grayscale, as:

$$\mathrm{PSNR}(\mathrm{G1},\mathrm{G2}) = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}(\mathrm{G1},\mathrm{G2})}\right) \ldots 1$$

where the Mean Square Error (MSE G1, G2) can be computed as:

$$\mathrm{MSE}\ (\mathrm{G1},\mathrm{G2}) = 1/\mathrm{nm}\sum_{\mathrm{i}=1}^{\mathrm{n}}\sum_{\mathrm{j}=1}^{\mathrm{m}}\left(\mathrm{G}_{1\mathrm{ij}} - \mathrm{G}_{2\mathrm{ij}}\right)^2 \ldots 2$$

The greater the PSNR value is, the better the image quality is, and hence a better embedding method is indicated.

SSIM can be computed as:

$$\mathrm{SSIM}(\mathrm{G1},\mathrm{G2}) = \mathrm{b}(\mathrm{G1},\mathrm{G2})\mathrm{c}(\mathrm{G1},\mathrm{G2})\mathrm{s}(\mathrm{G1},\mathrm{G2}) \ldots 3$$

where

$$\begin{cases} \mathrm{b}(\mathrm{G1},\mathrm{G2}) = 2\mu_{\mathrm{G1}}\mu_{\mathrm{G2}} + \mathrm{C}_1/{\mu_{\mathrm{G1}}}^2 + {\mu_{\mathrm{G2}}}^2 + \mathrm{C}_1 \\ \mathrm{c}(\mathrm{G1},\mathrm{G2}) = 2\sigma_{\mathrm{G1}}\sigma_{\mathrm{G2}} + \mathrm{C}_2/{\sigma_{\mathrm{G1}}}^2 + {\sigma_{\mathrm{G2}}}^2 + \mathrm{C}_2 \\ \mathrm{s}(\mathrm{G1},\mathrm{G2}) = 2\sigma_{\mathrm{G1G2}} + \mathrm{C}_3/\sigma_{\mathrm{G1}} + \sigma_{\mathrm{G2}} + \mathrm{C}_3 \end{cases}$$

where μG1, σG1, and σG1G2 represent the mean of G1, variance of G1, and covariance of G1 and G2, respectively. C1, C2, and C3 are small constants. SSIM value ranges from 0 to 1, where 0 means that there is no correlation between G1 and G2, whereas 1 means ideal coincide (G1 = G2).

Figure 14- Detection of the tampering areas. The first column represents the watermarked images, while the second column contains tampered versions of the images in the first column. Note the fake glasses around Lena's eyes, the gray camouflaged M letter in the right corner of the girl's photo, and the bracelet in the left hand of Barbara. The third column contains the same images with the tampered areas, indicated with black color.

Table 2- Performance measures of the proposed algorithm through the use of Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) for measuring image quality.

Image	PSNR	SSIM
Lena	44.2416	0.9949
Girl	46.5576	0.9956
Barbara	42.5110	0.9949

According to image terminologies, a PSNR with a value of 41 and higher is considered as an excellent rate. Hence, the PSNR of these standard images, achieved by using the proposed algorithm, reflects its efficiency. It also shows excellent SSIM values of higher than 0.99.

Table 3-The consumed time for process completion in seconds

Image	The embedding elapsed time	The extraction elapsed time
Lena	15.696159	16.399570
Girl	9.500429	10.904051
Barbara	10.422835	12.057025

Table 4-Performance comparison of the proposed scheme with previous image authentication schemes

Scheme	Average PSNR of Watermarked Image
Proposed	44.4367dB
Nguyen et al. [15]	40.58 dB
Hu et al. [16]	38.87 dB
Lo and Hu [17]	51.62 dB

It is noticed from Table-4 that the algorithm presented in this paper outperforms some other algorithms, except for Lo and Hu. The relative degradation in the PSNR of the proposed algorithm is reasonable, due to the fact that changing only the least significant bit in only one coefficient will propagate the change to a much wider region upon moving from the lowest wavelet level to the highest one during the application of the inverse transform. Thus, the PSNR of the watermarked image will certainly be affected.

Also, the IWT depends on the lifting scheme, which in its nature has to make some rounding for the floating point numbers during its operations to get the integer values, thus losing some precision for the resulted integer watermarked image pixel values. Fig. (15) below shows the IWT, its inverse operation (IIWT) for the 8*8 block "F" which is shown in fig. (15, a), and how the change in only one LSB of the second WT level causes some relative degradation in the retrieved block PSNR. Such that, if we change the greatest coefficient in "f2", which is shown in fig. (15, c), from 48 to 49 to have an even parity, then the inverse integer wavelet transform will give the "retrieved f1" which is shown in fig. (15, d) and the "retrieved f" as in Figure-(15, e).

F (original block) =

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	255	1	1	1	1
1	1	1	255	255	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	255
1	1	1	1	1	1	255	255

(a)

F1 (first level IWT approximation) =

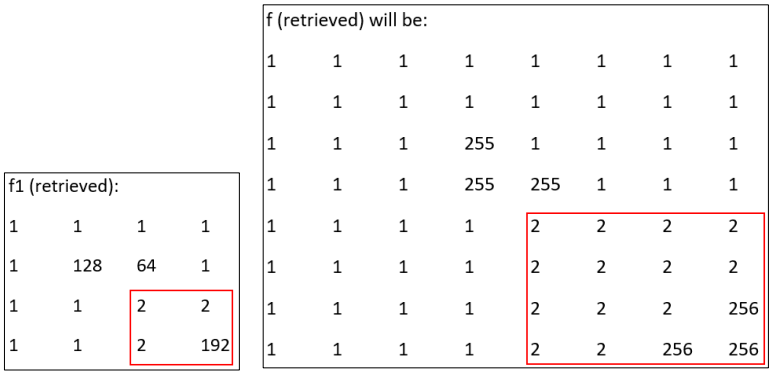
1	1	1	1
1	128	64	1
1	1	1	1
1	1	1	191

(b)

f2 (second level IWT approximation)

32	17
1	48

(c)



(d) (e)

Figure 15-The first row shows (a) an original block "F" of size 8*8 pixels, the approximation (low-low) "F1" of the first IWT level of "F", and the approximation "f2" of a second IWT applied to "F1". The second row shows (d) the "retrieved f1" after applying IIWT on "f2" and its complement second level IWT sub bands, and then (e) the "retrieved f" by applying another IIWT on the "retrieved f" and its complement first level IWT sub bands.

5. Conclusions

This paper presents an image watermarking technique based on IWT, which is more practical, faster, and less complicated than the traditional DWT. The DWT tends to produce floating point numbers; therefore, it needs complicated and some rounding operations. As the image pixels have integer values, the IWT is more suitable to conduct integer-to-integer operations directly. Making use of the low and mid frequency bands of the second wavelet transform, without using the first level bands, plays a good role in preserving a good image quality (measured by PSNR and SSIM). This is actually achieved due to the flexibility in the selection of the appropriate frequency level and bands, which the second level can offer more than the first level, can do. On the other hand, this paper utilizes the principle of the parity bit checking, which is very well-known and used widely around the world to investigate received signal correctness, making it a good candidate for much of the applications, especially in devices with a low computation capabilities. Therefore, this paper exploits parity bits checking as a way to investigate the authentication of the received images. Parity bit checking is used in this paper for its simplicity and low computation requirements, but unfortunately, as in the figures of the investigated images, it is noticeable that some tampered blocks are not identified correctly. This happens incidentally when all the addressed coefficients of the tampered block happen to have an even parity even after their values alteration. A more complicated, but much more powerful, hamming code approach can be utilized in the future to reduce the parity bit vulnerabilities. Nevertheless, compared with performance described by other articles, it is clear that the proposed approach still has interesting results.

References

1. Mohamed H. and Chunyan W. **2018**. Adaptive Image Self-Recovery Based on Feature Extraction in the DCT Domain. *IEEE Access Journal*, **6**: 67156-67165. DOI: 10.1109/ACCESS .2018. 2879404
2. Rajput V. and Ansari I. **2019**. Image tamper detection and self-recovery using multiple median watermarking. *Multimedia Tools and Applications*, https://doi.org/10.1007/s11042-019-07971-w.
3. Barania M., Valandara M. and Ayubib P. **2019**. A new digital image tamper detection algorithm based on integer wavelet transform and secured by encrypted authentication sequence with 3D quantum map. *Optik - International Journal for Light and Electron Optics*, **187**:205–222. https://doi.org/10.1016/j.ijleo.2019.04.074
4. Tai W. and Liao Z. **2018**. Image self-recovery with watermark self-embedding. *Signal Processing: Image Communication* **65**:11–25. https://doi.org/10.1016/j.image.2018.03.011
5. Belferdi W., Behloul A. and Noui L. **2018**. A Bayer pattern-based fragile watermarking scheme for color image tamper detection and restoration. *Multidim Syst Sign Process* **30**:1093–1112. https://doi.org/10.1007/s11045-018-0597-x

6. Abdelhakim A., Saleh H. and Abdelhakim M. **2019**. Fragile watermarking for image tamper detection and localization with effective recovery capability using K-means clustering. *Multimedia Tools and Applications*. doi.org/10.1007/s11042-019-07986-3.

7. Durgesh S., Singh S. **2016**. Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability. *J. Vis. Commun. Image* R. **38**: 775–789. https://doi.org/10.1016/j.jvcir.2016.04.023

8. Fang C., Bowen A., Jinwei W., Dengpan Y., Huili W. **2017**. Hierarchical recovery for tampered images based on watermark self-embedding. *Elsevier B.V.*, **46:** 52-60. https://doi.org/10.1016/j.displa.2017.01.001

9. Nguyen T., Chang C., Yang X. **2016**. A reversible image authentication scheme based on fragile watermarking in discrete wavelet transform domain. *International Journal of Electronics and Communications (AEÜ)*, **70**: 1055–1061. https://doi.org/10.1016/j.aeue.2016.05.003

10. Rhayma H., Makhloufi A., Hamam H. and Hamida A. **2019**. Semi-Fragile Self-Recovery Watermarking Scheme Based on Data Representation Through Combination. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-019-7244-x

11. Wu C. and Shih Y. **2013**. A Simple Image Tamper Detection and Recovery Based on Fragile Watermark with One Parity Section and Two Restoration Sections. *Optics and Photonics Journal*, **3**:103-107. doi:10.4236/opj.2013.32B026

12. Rakhmawati L., Wirawan W. and Suwadi S. **2019**. A recent survey of self-embedding fragile watermarking scheme for image authentication with recovery capability. *EURASIP Journal on Image and Video Processing* . https://doi.org/10.1186/s13640-019-0462-3

13. Zhang Z., Sun H., Gao S. and Jin S. **2018**. Self-recovery reversible image watermarking algorithm. *PLoS ONE*, **13**(6): e0199143. https://doi.org/ 10.1371/journal.pone.0199143.

14. Artru R. and Roux L. **2019**. Digital Watermarking Of Video Streams: Review Of The State-Of-The-Art. *arXiv:1908.02039v2 [eess.IV]*. https://arxiv.org/abs/1908.02039

15. Nguyen TS., Chang CC. and Chung TF. **2014**. A Tamper-Detection Scheme For Btc compressed Images With High-Quality Images. *KSII Trans Internet Inf Syst* , **8**(6). DOI: 10.3837/tiis. 2014. 06.011

16. Hu YC., Lo CC., Chen WL. And Wen CH. **2013**. Joint image coding and image authentication based on absolute moment block truncation coding. *J Electron Image*, **22**(1):1–12. DOI: 10.1117/1.JEI.22.1.013012

17. Lo CC. and Hu YC. **2014**. A novel reversible image authentication scheme for digital Images. *Signal Process*, **98**:174–85. https://doi.org/10.1016/j.sigpro.2013.11.028