

New Semi-Fragile Image Authentication Watermarking Techniques Using Random Bias and Non-Uniform Quantization

Kurato Maeno ^{*a}, Qibin Sun ^{**b}, Shih-Fu Chang ^{**b}, Masayuki Suto ^{*a}

^aOki Electric Industry Co., Ltd. Futaba-cho, Takasaki, Japan 370-8585
E-mail: maeno284@oki.com, suto627@oki.com

^bDepartment of Electrical Engineering, Columbia University, New York, NY 10027, USA
E-mail: qibin@ee.columbia.edu, sfchang@ee.columbia.edu

ABSTRACT

Semi-fragile watermarking methods aim at detecting unacceptable image manipulations, while allowing acceptable manipulations such as lossy compression. In this paper, we propose new semi-fragile authentication watermarking techniques using random bias and non-uniform quantization, to improve the performance of the methods proposed by Lin and Chang [1]. Specifically, the objective is to improve the performance tradeoff between the alteration detection sensitivity and the false detection rate.

Keywords: Semi-fragile watermarking, Wavelet Transform, Random Bias, Non-uniform Quantization, JPEG2000

1. INTRODUCTION

To authenticate the content of a given image securely and transparently, there are usually two integrated procedures: content-based signature generation and content-based watermarking.

Semi-fragile watermarking methods aim at detecting unacceptable image manipulations, while allowing acceptable manipulations such as lossy compression. This paper proposes new techniques to improve the performance of semi-fragile authentication watermarking methods proposed by Lin and Chang [1]. Specifically, the objective is to improve the performance tradeoff between the alteration detection sensitivity and the false detection rate.

In the original method, each signature bit is generated by comparing two DCT coefficients belonging to the same frequency but from different blocks. Multiple signature bits are generated by randomly selecting a sequence of pairs of DCT coefficients from the image. The random selection function is dependent on the user key.

To address the noise caused by practical implementations, the technique introduced an error tolerance margin to reduce the false alarm rate. But the use of such error margins may also lead to missing some malicious alterations of images. One example of such content altering operation is smoothing attack (e.g., delete objects by filling with background colors / textures) in which the changes to the DCT coefficient difference may be within the tolerance margin and thus the attack cannot be detected. In experimenting with several other content altering manipulations (e.g., masking, delete contents, etc.), we also observed similar phenomena.

The above issue is attributed to two sources of limitation in the original technique. First, due to the requirement in controlling the signature length, the relationship of two DCT coefficients in the same pair is encoded by 1 single bit only. Second, relationships among the DCT coefficients in a local proximity area are not explored. In this paper, we addressed these two problems and proposed two techniques to improve the performance of semi-fragile authentication watermarking. In addition, we extend the JPEG DCT-based watermarking technique to the wavelet domain and extend the acceptable compression to JPEG 2000.

In the first method, we explore the correlation among coefficients in a local window. An interesting phenomenon shown in experiments indicated a given manipulation tends to cause similar change patterns to coefficients in a local window. Such similar patterns result in a clustered distribution in the (original difference – new difference) plane. The fixed encoding boundary used in the original technique has a potential issue of missing all pairs of coefficients for an attack. In this new method, we introduce a novel component which adds a random bias factor to the decision boundary. Such randomization factor spreads out the coverage of each signature bit in catching the malicious attack.

In the second method, we propose a non-uniform quantization scheme which uses multi-bit non-uniform quantizer to encode the transform coefficient difference in each pair, and uses different quantizers in the signature verification site.

We use multiple bits to improve the accuracy in encoding the relationships between paired transform coefficients. We use non-uniform quantizers to explore the non-linear mapping between the coefficient differences in the original image and the compressed image. After thoroughly analyzing the properties of different distortions caused by acceptable (e.g., lossy compression) and unacceptable (e.g. copy-paste) manipulations, we use a non-uniform quantizer to generate the raw signature, which is comprised of several bits instead of one single bit, from the difference between two wavelet coefficients. The non-uniform quantization consists of two steps. First, we use different quantization step sizes to quantize the difference between these two selected coefficients according to the magnitude of the difference. Second, we assign different quantization step sizes for the signature generator and the signature verifier which is guided by the observations on the difference of distortion properties from different manipulations especially on lossy compression.

Obviously these raw signatures require much more capacities for signature storage as well as for watermark embedding. Therefore, for this method, we also propose a codeword table-based solution to shorten the length of the generated signature.

2. PREVIOUS WORK ON THE DCT-BASED SEMI-FRAGILE WATERMARKING

In this section, we review the main approach to achieve the semi-fragile watermarking proposed by Lin and Chang [1]. This unique semi-fragile authentication watermarking technique is well recognized for its capability of providing deterministic guarantee of zero false alarm rate and statistical guarantee of miss rate in distinguishing malicious attacks from JPEG lossy compression. The authors have deployed a popular software that's freely downloadable and available for testing from an online web site [7].

When an image is compressed with JPEG, its image pixels are transformed to DCT coefficients, and then quantized. Lin and Chang found that the magnitude relationship between two coefficients remains invariable through repetitive JPEG compression. They demonstrated that semi-fragile image authentication for JPEG is feasible using this property [1]. In their method, the authenticity of the image is verified by 1-bit signature bit which represents the magnitude relationship between two DCT coefficients.

Procedures of Signature Generation:

In the signature generation site, a signature bit is generated for two coefficients corresponding to the same frequency in two different DCT blocks, which are selected using a "pairing vector" generated by a pseudo-random number generator. Given a pair of coefficients (q_i and p_i) from these two blocks, equation 1 is applied.

$$Sig_i = \begin{cases} 0 & (p_i - q_i \geq 0) \\ 1 & (p_i - q_i < 0) \end{cases} \quad (1)$$

where

p_i, q_i : DCT transformed coefficients in the same frequency location from two different blocks, and locations of p_i, q_i are determined by vector v_i , $Location(p_i) = Location(q_i) + v_i$
 Sig_i : Signature bit for the relationship between p_i and q_i

Then signature bits are embedded into other DCT coefficients which can be selected using another pseudo-random sequence. For the details of watermark embedding, readers are referred to [1].

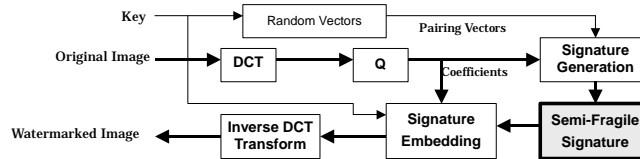


Figure 1. Semi-fragile watermark generation steps

Procedures of Signature Verification:

At the verification site, as with the embedding site, DCT coefficients are verified by signature bits using a similar manner as the signature generation site. The verification procedure consists of three steps: 1) extracting signatures that have been embedded by the embedding site, 2) generating difference values from DCT coefficients, and 3) verifying the extracted signatures and the generated difference values according to three conditions 1, 2 and 3 listed below. Condition 2 is placed here to allow a certain noise tolerance margin. To tolerate the noise introduced by some benign manipulations, such as color transforms, different codec implementations, and integer rounding. A relationship that satisfies any one of

these 1 - 3 conditions shall be considered as not being manipulated, otherwise the coefficient pairs are considered as manipulated.

$$\begin{cases} p'_i - q'_i > M & \text{and} & Sig_i = 0 & \text{(condition1)} \\ |p'_i - q'_i| \leq M & \text{(don't care for } Sig_i) & & \text{(condition2)} \\ p'_i - q'_i < -M & \text{and} & Sig_i = 1 & \text{(condition3)} \end{cases}$$

where

p'_i, q'_i : DCT transformed coefficients which are used to generate signature bit “ Sig_i ” at the verification site (typically, after lossy compression), and locations of p'_i, q'_i are determined by vector v_i (same as generator)
 $Location(p'_i) = Location(q'_i) + v_i$

M : Margin value to avoid the false alarm caused by lossy compression using different quantizers or noise introduced by different implementations

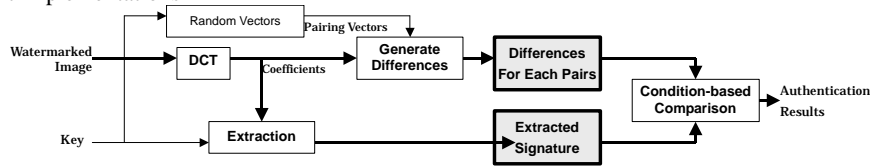


Figure 2. Semi-fragile watermark verification steps

As the result of the above, a pair of coefficients which falls into $p_i - q_i \geq 0$ at the signature generation site shall be considered as being manipulated if it falls into $p'_i - q'_i < M$ (the size relationship is reversed) at the verification site. Similarly, one which falls into $p_i - q_i < 0$ at the signature generation site shall be considered as being manipulated if it falls into $p'_i - q'_i > -M$ (the size relationship is reversed) at the verification site.

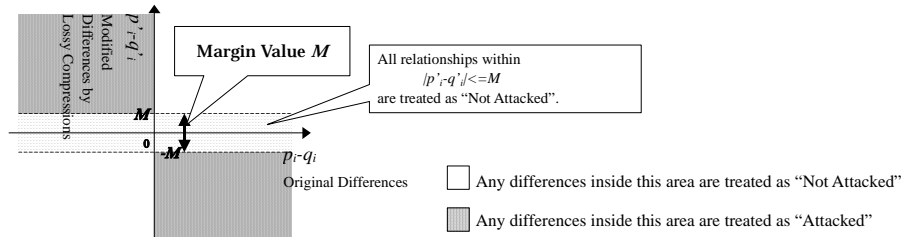


Figure 3. Differences map for attack detection

Here, a big security hole may arise from the relationships which fall into $|p'_i - q'_i| \leq M$ (condition 2) that is placed here to allow for some noises at the verification site, because they are not considered as being manipulated regardless of the size relationships at the signature generation site. It means that, if an attacker manipulates the image, making the absolute value of the difference between p'_i and q'_i below Margin Value M , this attack will never be detected whatever the coefficient values p_i and q_i (and v_i) at signature generation are (meshed area in Figure 3 above).

If this type of attack is practically meaningless, impossible and/or very difficult to achieve, this problem may be negligible. However, in fact, it is very easy to achieve and even could be very harmful to certain contents. For example:

- Deletion of an object
 Objects can be deleted very easily especially for images with a very homogeneous background such as a document image. In the other cases, objects can also be deleted by pasting smooth textural background over them.
- Insert of an object
 This can be done by drawing a very light-colored object on a background.

Neglecting these attacks may cause extremely harmful defects especially for digital watermarking which should prevent evidential images and document images from being manipulated.

Now, we propose two solutions to overcome these defects and further improve the alteration detection rate: 1) Random Bias method and 2) Non-uniform quantization method. Then we compare them to Lin and Chang's method in the wavelet

domain. In the rest of the paper, we discuss our techniques in the context of wavelet transform of JPEG2000, though the techniques can be applied to block-based transforms like DCT of JPEG also.

3. WATERMARKING ALGORITHM

In this section, we propose 1) Random Bias method, and 2) Non-uniform Quantization method. Details are described in subsequent sections. 1) Random Bias Method makes it difficult for attackers to keep the difference below the margin value M by adding a random bias to the difference between two coefficients p_i and q_i . 2) Non-uniform Quantization method firstly removes “don’t care” parts introduced by the margin value by using multi-bit representation for signature generation, and then reduces the degradation of image visual quality caused by long signature embedding by shortening signature bits with the codeword assignment table, while keeping the high detection rate.

3.1. Random Bias Method

First, we observed how pair of coefficients were affected by manipulations in the wavelet domain (see **Figure 4**). We found that manipulations involve many different effects and have common features discussed below at the same time in many cases.

- The relationships between two coefficients which have been manipulated result in a certain clustered distribution on the (original difference-new difference) plane.
- Relationships between two coefficients gather around 0 if the attack such as object deletion occurs. This can be illustrated by the graph shown in **Figure 4(a)**.

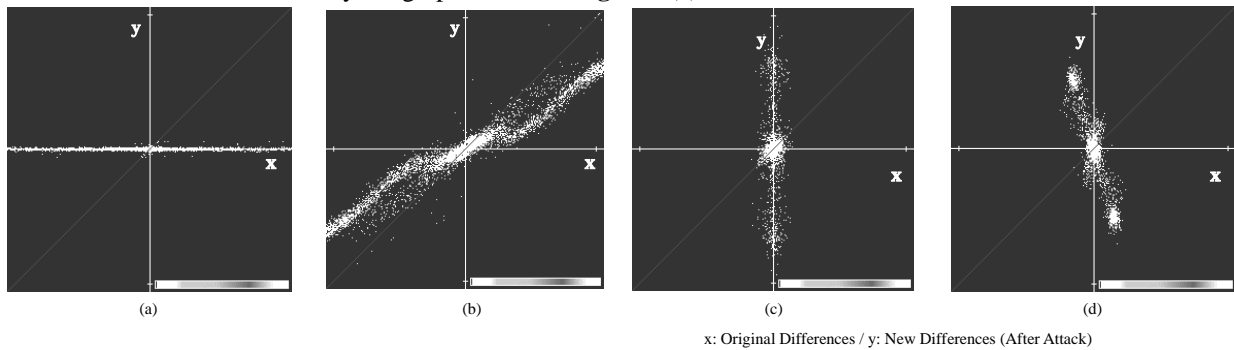


Figure 4. Distribution of coefficient difference before and after manipulation: (a) Delete Image Objects, (b) Change Luminance (255->99), (c) Delete Image Objects, (d) Change Hue (Results of a color component), Coefficients belong to 1LL subband transformed by 5x3 integer Wavelet filter (JPEG2000 Compatible).

If the relationships between two coefficients which have been manipulated don't lead a clustered distribution, shifting the thresholds in Lin and Chang's method from 0 might decrease the alteration detection performance, because many relationships change around 0 when manipulated (see **Figure 5**). In this case, in order to prevent the detection rate drop, the signature length should be extended and multiple thresholds must be used to verify the difference between a pair of coefficients. In many cases, however, manipulating relationships results in a cluster. Therefore, verifying more than one relationships within the cluster with different thresholds (i.e. **Figure 6**) will catch manipulations those are so far not detectable using a fixed zero-value threshold and decrease the possibility of misses.

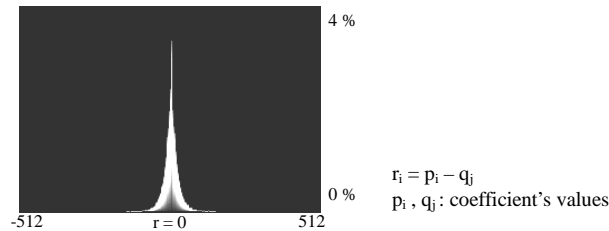


Figure 5. Histogram of differences distributions for natural image under attack manipulations

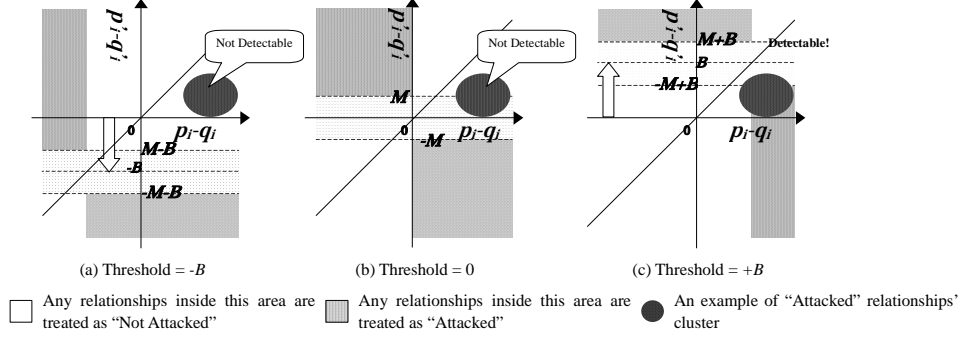


Figure 6. Examples of various thresholds: Note the attack becomes detectable after the threshold value is changed from 0 to B.

Here we explain our proposed random bias method. Random bias method adds random values to the difference between two coefficients before the difference is encoded to the signature bit. The detailed algorithm of random bias method is as follows.

Procedure of Signature Generation:

$$Sig_i = \begin{cases} 0 & (p_i - q_i + B_i \geq 0) \\ 1 & (p_i - q_i + B_i < 0) \end{cases} \quad (2)$$

where

- p_i, q_i : Wavelet transformed coefficients in the same Subband, and locations of p_i, q_i are determined by vector v_i
- $Location(p_i) = Location(q_i) + v_i$
- B_i : the i th element of pseudo-random number sequence \mathbf{B} as random bias
- Sig_i : Signature bit for the relationship between p_i and q_i

Procedure of Signature Verification:

A pair of coefficients which meets any one of 4 – 6 conditions below shall be considered as not being manipulated, otherwise considered as being manipulated.

$$\begin{cases} p'_i - q'_i + B_i > M & \text{and} & Sig_i = 0 & \text{(condition4)} \\ |p'_i - q'_i + B_i| \leq M & & \text{(don't care for } Sig_i) & \text{(condition5)} \\ p'_i - q'_i + B_i < -M & \text{and} & Sig_i = 1 & \text{(condition6)} \end{cases}$$

where

- p'_i, q'_i : Wavelet transformed coefficients which are used to generate signature bit “ Sig_i ” at the verification site (typically, after lossy compression), and, and locations of p'_i, q'_i are determined by vector v_i (same as generator)
- $Location(p'_i) = Location(q'_i) + v_i$
- B_i : The i th element of pseudo-random number sequence \mathbf{B} as random bias (same sequence of generator)
- M : Margin value to avoid the false alarm caused by noises introduced by acceptable manipulations

Again, with the original method, manipulation is detected by comparing two coefficients in terms of their difference values. Adding a bias here shifts the basis for comparing difference value, from zero to the selected bias value. We expect that shifting biases randomly will enable detection of the alterations that have been undetectable so far, leading to an increased detection rate. For example, as shown in **Figure 6**, differences (before and after the attack) of coefficient pairs are concentrated in a cluster. If we use a fixed threshold (0), none of the coefficient pair will be detected. By randomly changing the threshold for each coefficient pair, it’s reasonable to expect some coefficient pairs will be detected when the threshold is shifted to a positive bias.

3.2. Non-uniform Quantization Method

Non-uniform Quantization method consists of two steps; the first step is to generate raw signatures from the difference between a pair of coefficients p_i and q_i , by quantization. Unlike the previous approach, each of these raw signatures is represented by multiple bits. The second step is to concatenate a certain number of pairs of these raw signatures to

produce one new signature, then shorten it by hashing to make the average representation of whole signature 1 bit per one pair as the same as previous approaches.

There are two reasons why we call this method “non-uniform quantization”; the first reason is to change the quantization step sizes depend on the magnitude of the difference value. The second one is that the signature verification site will use quantization step sizes differing from those used at the signature generation site.

Analysis of Difference Changes by Lossy Compressions

In this section we explain our observation of how a pair of coefficients is affected by lossy compression in the wavelet domain. **Figure 7** shows how the difference value of a pair of coefficients changes when a natural image is lossy-compressed (JPEG2000). It plots on a r_i - r'_i plane with two difference values (r_i, r'_i) obtained respectively from the identical points of two images (the original and the lossy-compressed). The x-axis indicates the difference values (r_i) obtained from the original image, and the y-axis indicates the difference values (r'_i) modified by the JPEG2000 codec. The mean curve and the standard deviation curve indicate the overall distribution of the mean value and standard deviation respectively calculated based on the difference value from the original image.

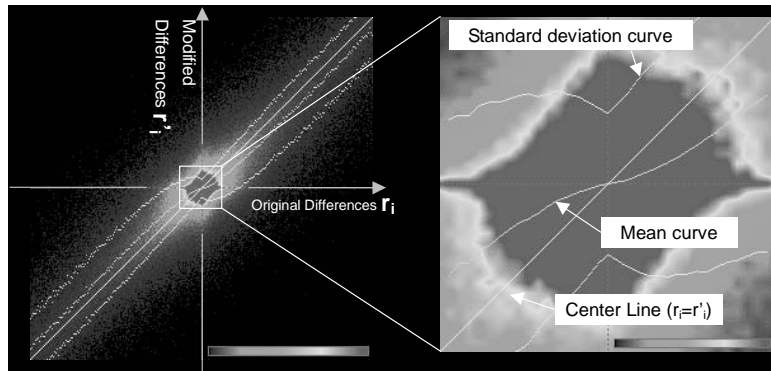


Figure 7. Differences changes by JPEG2000 lossy compressions: Original differences $r_i = p_i - q_i$. Modified differences $r'_i = p'_i - q'_i$. p_i, q_i, p'_i, q'_i are wavelet coefficients of 1LL using 5x3 wavelet filter and reversible color transform. p'_i, q'_i are modified by JPEG2000 lossy compression using 9x7 wavelet filter and irreversible color transform with 5 decomposition level and 0.25bpp. The coefficients of (p_i, p'_i) are coefficients of same location. (q_i, q'_i) are also same.

As a result of observation, we found the following features in many cases.

- After lossy compression, the difference value changes to a smaller value when it is positive while it changes to a bigger value when negative. (it gets closer to 0 in both cases)
- The difference value changes considerably around the value 0 as contrasted with other values (although the absolute variance value is small)

Images compressed by other lossy compression such as JPEG2000/ JPEG, showed much the same distributions in greater or lesser degrees. In addition, the above observations hold for a wide range of image types, such as document image, natural image and computer graphics image.

Here consider the reasons for these phenomena. We may say that lossy compression is a kind of low-pass filtering since it intends to diminish the energy of image data as it contains much high frequency elements. The image will be smoothed more and the difference of wavelet coefficients will be smaller when the low-pass filtering is applied to the image. This is the most likely reason that absolute values of differences of many coefficients become smaller than the originals. Additionally, a noise called ringing effect in JPEG2000, or mosquito noise in JPEG, may appear near the border of the image. These noises make coefficient values near the border fluctuate, and therefore make difference values fluctuate too. This seems to be another reason for causing some variations in distribution.

From observation described above, it seems that the possibility of a false alarm may decrease while the detection efficiency increases if we generalize the above observations and assume that 1) The magnitude of difference value around 0 are changed bigger than others, and 2) magnitude of difference value at the signature verification site is smaller on average than that at the signature generation site.

Non-uniform Quantization Method

Now we explain non-uniform quantization method which is developed basing on the above observations and hypothesis. **Figure 8** and **Figure 10** are block diagrams of non-uniform quantization method. The modules through “Differences Generation” block are the same as Lin and Chang’s method and the random bias method described earlier. The unique functions of this method are accomplished by subsequent blocks.

Here we describe the signature generation procedure. The input image gets wavelet coefficients after color transform and wavelet transform as with the case of the random bias method. The vector obtained from “Random Vector generator” block generates pairs of wavelet coefficients and calculates differences from each pair (the process so far is the same as random bias method and the original Lin and Chang’s method).

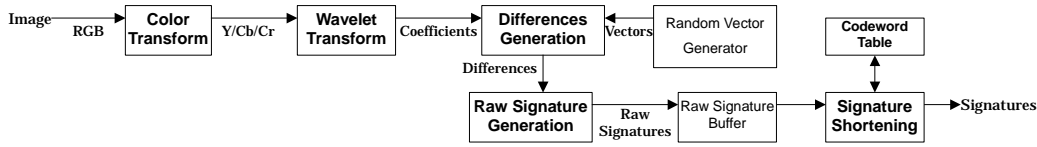


Figure 8. Signature Generator Block Diagram

Table 1. Raw Signatures Generation Table

Difference Range	Raw Signature
$Q_1 < p_i - q_i$	0
$-Q_1 < p_i - q_i \leq Q_1$	1
$p_i - q_i < -Q_1$	2

p_i, q_i : Wavelet transformed coefficients in the same Subband, and locations of p_i, q_i are determined by vector v_i
 $\text{Location}(p_i) = \text{Location}(q_i) + v_i$, Q_1 : Quantization step size as threshold

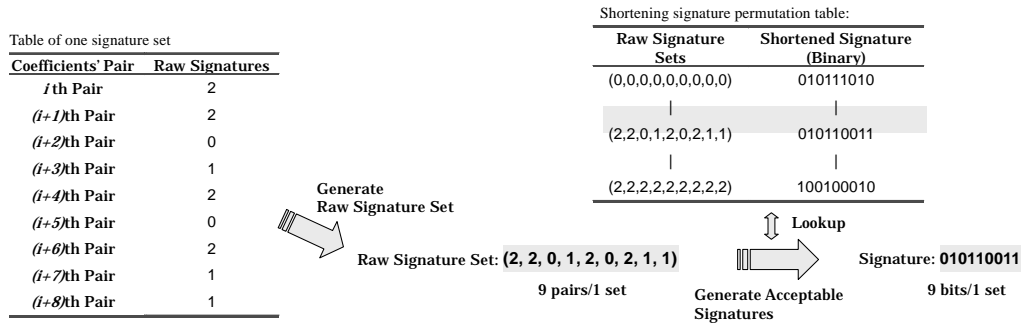


Figure 9. Examples of Signature Generation

Non-uniform quantization method generates raw signatures which use several bits for each coefficient pair while Lin and Chang’s method generates one-bit signature for each. The example of generating a raw signature with multiple bits is shown in **Table 1** (in this example, a signature which takes 3 values (2bits) is generated for each pair).

The signature verification procedure will be a little more complicated than the generation procedure. The procedure until acquisition of the difference from a pair of coefficients is the same as the generation site. “Non-uniform quantize” block generates all acceptable raw signatures, depending on the difference value according to the rules described in **Table 2**. Acceptable raw signature means the signature value which should be generated by the signature generation site for the difference value that is obtained at the signature verification site. For example, if ‘1’ is generated at the signature generation site, it is considered as not being manipulated at the signature verification site when the difference value computed at the verification site is within the range of $-Q'_2 < p'_i - q'_i \leq Q'_2$.

It's important to understand the acceptance rules listed in **Table 2**. As shown in **Figure 12**, the acceptable region in the “new difference-old difference” plane is more complicated than the simpler one for the original method (shown in **Figure 3**). Here, multiple parameters, Q_1, Q'_1, Q'_2 can be used to control the acceptable regions and match them to the distributions observed in typical acceptable manipulations (see **Figure 7**).

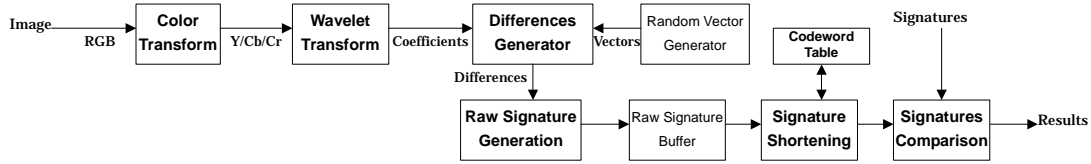


Figure 10. Signature Verifier Block Diagram

Table 2. Acceptable Raw Signatures Generation Table

Difference range	Acceptable Raw Signatures
$Q'_2 < p'_i - q'_i$	0
$Q_1 < p'_i - q'_i \leq Q'_2$	0 / 1
$ p'_i - q'_i \leq Q_1$	1
$-Q_2 \leq p'_i - q'_i < -Q_1$	1 / 2
$p'_i - q'_i < -Q'_2$	2

p'_i, q'_i : Wavelet transformed coefficients in the same Subband, and locations of p'_i, q'_i are determined by vector v_i
 $\text{Location}(p'_i) = \text{Location}(q'_i) + v_i, Q_1, Q'_2$: quantization step sizes as thresholds

Table of one acceptable signature set

Coefficients' Pair	Acceptable Raw Signatures
i th Pair	1 / 2
$(i+1)$ th Pair	2
$(i+2)$ th Pair	0 / 1
$(i+3)$ th Pair	1
$(i+4)$ th Pair	2 / 1
$(i+5)$ th Pair	0
$(i+6)$ th Pair	2
$(i+7)$ th Pair	1
$(i+8)$ th Pair	1

Shortening signature permutation table:

Raw Signature Sets	Shortened Signature (Binary)
(0,0,0,0,0,0,0,0)	010111010
(2,2,0,1,2,0,2,1,1)	010110011
(2,2,2,2,2,2,2,2)	100100010

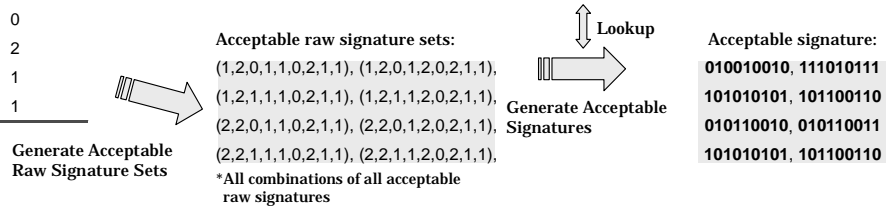


Figure 11. Example of acceptable signatures generation

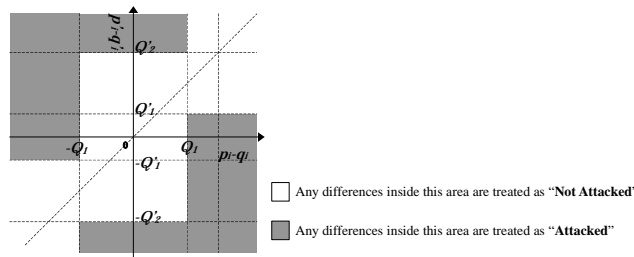


Figure 12. Attack Detect Area for Non-uniform Quantize Solution

“Signature Shortening” block is also the same as in the signature generation site in that it generates one signature by concatenating a certain number of raw signatures except that there are more than one raw signatures acceptable while signature generation site has only one raw signature for one pair. Consequently, the verification site generates an acceptable raw signature set from the combination of all acceptable raw signatures. And then, it generates binary signatures for each raw signature vector in the set by the same procedure as the signature generation site (see **Figure 11**). “Signature Comparison” block compares the generated acceptable signature with the one generated at the signature generation site. If the signatures of the verification site don’t include that signature of the generation site, it is considered as being manipulated, otherwise it is considered as not being manipulated. Consequently, we can expect high detection accuracy and obtain semi-fragile signature of average of 1 bit per one pair as with the case of the original method.

Codeword Assignment Table Generation

When shortening signatures, a raw signature set is used as an index to refer to the codeword assignment table, and the entry (one binary vector for each raw signature set) referred to is output. Since the number of possible raw signature sets far exceeds that of shortened binary signatures, collisions occur when different raw signatures refer to the same shortened binary signature. And this may cause misses in detecting alterations. However, since we know the likelihood distributions of the raw signatures, we can optimize the codeword assignment table to minimize the codeword collision probability mentioned above.

Raw signatures around the center ($|p_i - q_i| \leq Q_1$) take the highest probability of appearances. Therefore, raw signature sets consisting of the raw signatures at the center (in **Table 1, Table 2**). (1, 1, 1, 1, 1, 1, 1, 1, 1) has the highest probability. The farther it goes away from the center, the lower the probability of occurrence of raw signature sets is.

Figure 13 shows the relationships between the probability of occurrence and the raw signature’s distances from the center. Note that the distance D from the center to raw signature set $A(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9)$ is calculated as follows. The probability is obtained by empirical simulations using real images (similar to that shown in **Figure 5**) and is based on the assumption of independence among coefficient pairs.

$$D = \sum_{i=1}^9 |a_i - 1| \tag{3}$$

From these results, we can see that it takes the overwhelmingly high probability (50%-99%) when $D=0$. And the probability of $D=0$ for LL subbands and low frequency elements are lower than others. So, we can expect that optimizing the probability of collisions based on the probability of appearances will improve the detection accuracy. For example, if we set the $D=0$ table entries for no collisions, 50% of all signatures will be collision free for coefficients in the 2LL subband. If we adapt it to 1LH of U component, 99% of all will be collision free.

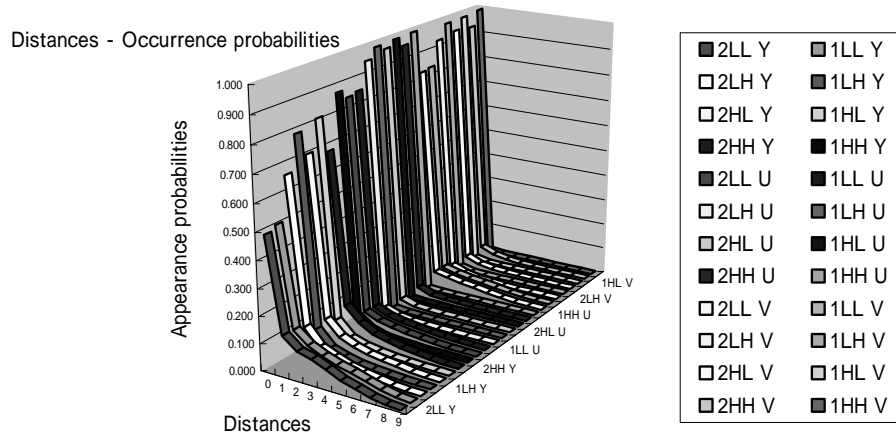


Figure 13. Relationships between distances and the probabilities of occurrence

Given the occurrence probabilities, the total number of input symbols, and the codeword length (e.g., 9 bits), one could obtain the optimal codeword assignment following a procedure similar to that used in the Huffman Code. Here, to simplify the testing, we categorize the raw signature sets to 3 groups depending on the distance D . The 1st group consists

of raw signature sets of $D=0$, the 2nd group contains raw signature sets of $0<D<4$, and the 3rd group contains others ($D \geq 4$) (See **Table 3**).

In the 1st group, there exists only (1,1,1,1,1,1,1,1), therefore, assigning only one shortened signature to this group will eliminate collisions. However, if we assign 1 bit per one pair, then we can only use a 9-bit (512 patterns) shortened signature per raw signature set. There are 19683 patterns for all raw signature sets and the codeword assignment table has 19683 entries. In each entry, one pattern out of 512 signatures is recorded. Consequently, it is obvious there exists collision because the identical signature value is used for more than one entries.

Here we assign the minimum number of binary signatures to the 3rd group which has the lowest probability of occurrence while we assign 280 binary signatures to the 2nd group to which 834 raw signature sets belong.

On the average there exist 2-3 raw signature sets that have the identical signature in 2nd group. Also, 231 binary signatures are assigned to 18848 raw signatures sets in the 3rd group with the lowest probability of occurrence. In this case, there exist approximately 80 raw signature sets that take the identical signature.

Table 3. Distance Groups and Collisions

Group	Distance D	The number of raw signature sets in the group	The number of shortened signatures in the group	The collisions rate
1	$D=0$	1	1	0% (no collision)
2	$0<D<4$	834	280	66.4%
3	$D \geq 4$	18848	231	98.8%

When forming the codeword assignment table in practice, the pseudo-random sequence randomizes index values, shortened signatures and overlapping patterns in each group. It also divides the image into blocks, use a random assignment in each group for each block in order to achieve an increased security.

The method described above will minimize miss probability resulting from shortening signatures while achieving the goal of shortening the signature length to 1 bit per one pair.

4. EXPERIMENTAL RESULTS

We have tested above-mentioned three methods with various images. Manipulations of different types were simulated, including the following

- | | | |
|--|----------------------------------|---------------------------|
| A. Delete (fill background textures) | B. Delete Background textures | C. Add a line drawing |
| D. Delete (fill background textures) | E. Paste another contents | F. Desaturate |
| G. Change Hue | H. Delete | I. Move |
| J. Replace by computer generated texts | K. Delete light colored contents | L. Delete guts |
| M. Add Grip | N. Skew | P. Delete papers contents |
| Q. Copy | R. Desaturate | S. Copy |

Figure 14 shows the block diagram for tests of random bias method and non-uniform quantization method. In each diagram, the upper row and the lower row represent the signature generation site and the signature verification site respectively and each generates the random vector, the random bias and the codeword assignment table by the same random sequence. The test pattern for image manipulation is as follows: the image is first manipulated and then lossy-compressed with JPEG2000 VM8.6 (JPEG2000 compression parameters are listed in **Table 4**) as shown in **Figure 15**. The parameters listed in **Table 5** are used for the wavelet transform in test blocks of **Figure 14**.

As a result, with random bias method, almost all manipulation areas including cropped parts which cannot be detected by the original Lin and Chang's method (area A, D and H of **Figure 15**) were successfully detected. The number of coefficients detected increased about 34% in total compared to the case with Lin and Chang's method. The detection rate for the areas where the luminance/color levels were slightly changed (area F of **Figure 15**) was low in all algorithms. Finally, we could not detect the area where the background was deleted (area B) because it was very small change (from white colored background to pure white (255) background).

As for non-uniform quantization method, its detection rate is nearly equal to the random bias method, which is increased about 35% compared to Lin and Chang's.

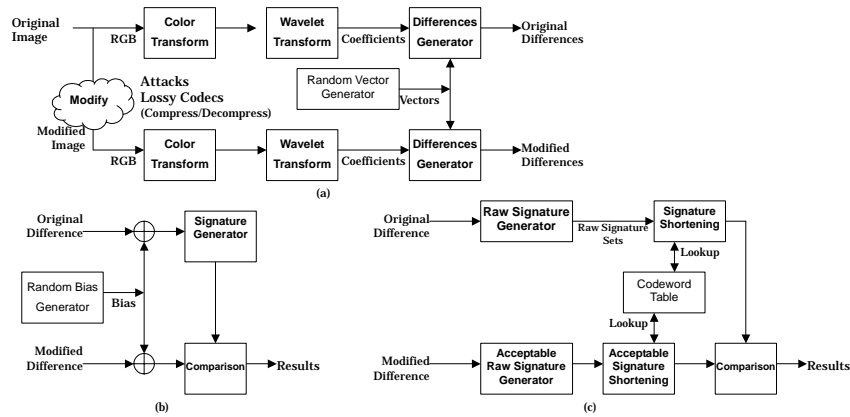


Figure 14. Block Diagrams: (a) Differences Generation Steps, (b) Random Bias Method, (c) Non-uniform Quantization Method

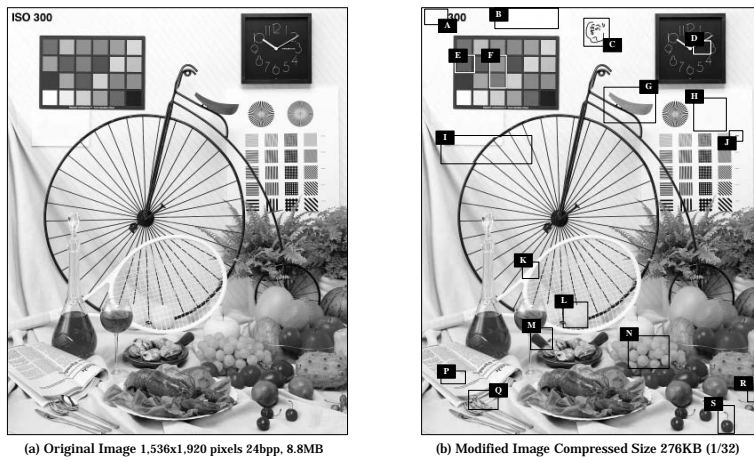


Figure 15. Test Patterns for Natural Image

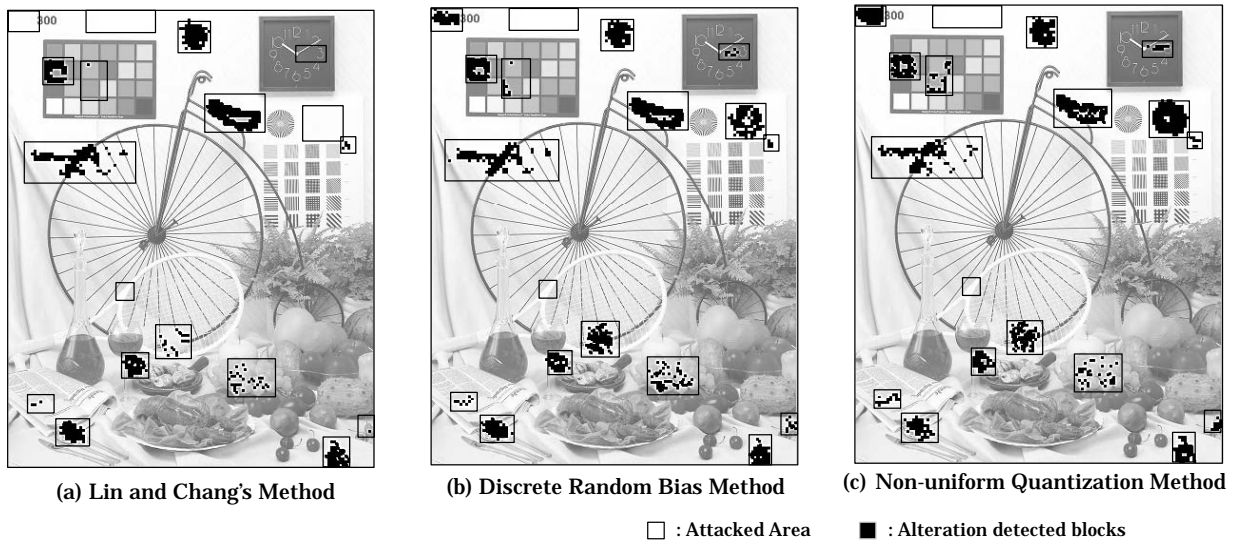


Figure 16. Experimental Results (a) Lin and Chang's Method ($M=25$) detects 687 blocks of **CEGIJLMNPQRS**, (b) Random Bias Method (Bias: $-28 / 0 / 28$, $M=19$) detects 919 blocks of **ACDEGHIJLMNPQRS**, (c) Non-uniform Quantization Method ($Q_1=30$, $Q_1'=2$, $Q_2'=60$) detects 930 blocks of **ACDEGHIJLMNPQRS**

Table 4. Conditions for Lossy Compressions

Compress options	Values
Lossy Compressor	JPEG2000 VM8.6
Wavelet Transform	9x7 Floating Point
Color Transform	Irreversible YCbCr
Tile size	128 x 128 (pixels)
Decomposition Level	5
Bitrate	0.75bpp (PSNR 31.89dB)

Table 5. Conditions for Testing

Testing options	Values
Wavelet Transform	5x3 Integer
Color Transform	Reversible (RCT)
Tile size	256 x 256 (pixels)
Testing Subband	2LL
Note - Select Parameters to guarantee no false alarm	

5. CONCLUSIONS

In this paper, we addressed the problems of a popular semi-fragile authentication watermarking technique and proposed new signature generation/verification algorithms. Our techniques improve the alteration detection sensitivity by analyzing and observing impact by various image manipulations on the transform coefficients and their relationships. Furthermore, we apply these algorithms to the coefficients that have been wavelet-transformed and, within the scope of the experiments conducted, proved that we can detect image manipulations even after JPEG lossy compression with different filters, and that no false alarm occurs while keeping the high detection sensitivity even for the object cropping attack.

In conclusion, our new algorithms demonstrate very encouraging performance for detecting various image attacks (including object cropping), even for images with a very homogeneous background such as a document image. In the field where the very strict image authenticity is strongly required, for example document images, it can be combined with the fragile watermarking to satisfy such strict requirements (our methods allow the fragile watermarking be considered as acceptable operation). In this case, the fragile watermarking can be used to ensure the whole image authenticity while the semi-fragile watermarking can be used to locate the altered points when whole image authentication fails. For authentication, the advantages of our methods are: 1) it can locate the altered points even if the altered image has been lossy-compressed, and 2) it allows flexible specification of the level of acceptable manipulation by setting a comparison threshold value. Our methods, like the original Lin and Chang's method, can be effectively used as an externally stored image signature, rather than embedded image watermarks.

Our future work will address lossy compression with different wavelet transform filters, study the alteration detection sensibility when image size changes, and more extensive test using more images of various types.

ACKNOWLEDGEMENTS

We appreciate very much the help provided by the Dr. Ching-Yung Lin in providing comments and advise during our research.

REFERENCES

1. C.-Y. Lin and S.-F. Chang, "Semi-fragile watermarking for authenticating JPEG visual content", *Proc. SPIE, Security and Watermarking of Multimedia Contents*, San Jose, California, pp.140-151, January 2000
2. S.Bhattacharjee and M.Kutter, "Compression Tolerant Image Authentication", *IEEE International Conf. On Image Processing*, Chicago, Oct 1998
3. M.Yeung and F.Mintzer, "An Invisible Watermarking Technique for Image Verification", *IEEE Proc. of ICMP*, Santa Barbara, Oct 1997.
4. M.Wu and B.Liu, "Watermarking for Image Authentication", *IEEE Proc. of ICIP*, Chicago, Oct 1998.
5. J. Fridrich, "Image Watermarking for Tamper Detection" *IEEE Int. Conf. On Image Processing*, Chicago, Oct 1998
6. M.P.Queluz, "Content-based Integrity Protection of Digital Images", *SPIE Conf. on Security and Watermarking of Multimedia Contents*, Vol.3657, pp.85-93, San Jose, January 1999
7. Self Authentication and Recovery Image (SARI) software and testing site. <http://www.ctr.columbia.edu/sari/>
8. JSA, "Graphic technology - Prepress digital data exchange - Standard colour image data(SCID) ISO/JIS-SCID", ISO, JSA, M.Kaji, 1995