

## SEMI FRAGILE WATERMARK WITH SELF AUTHENTICATION AND SELF RECOVERY

Woo Chaw Seng<sup>1</sup>, Jiang Du<sup>2</sup>, Binh Pham<sup>3</sup>

<sup>1</sup>Faculty of Computer Science and Information Technology,  
University of Malaya, 50603 Kuala Lumpur, Malaysia.  
<sup>2,3</sup>Queensland University of Technology, Brisbane, Australia.  
Email: cswoo@um.edu.my<sup>1</sup>, b.pham@qut.edu.au<sup>3</sup>

### ABSTRACT

*Robust watermarks are suitable for copyright protection in a DRM scenario. On the other hand, fragile watermarks are good for tamper detection applications. Semi fragile watermarks possess some properties of both robust and fragile watermarks at a moderate level. The need for semi fragile watermarks arises from the requirements of content authentication where the watermark must highlight malicious attacks while tolerating legitimate changes that do not alter the content severely. Very few watermarking scheme has both self authentication and self recovery features. We developed and evaluated a semi fragile watermarking scheme that offers these features. The scheme embeds a downscaled version of an image into the image's discrete wavelet transform subbands.*

*Our scheme provides content authentication by allowing high quality JPEG compression, minor local distortion, and minimal noise insertion. Other changes such as histogram equalisation, cropping, rotation, and mean filtering are classified as malicious attacks because it affects the visual quality of the image. The scheme is practical because it does not require a reference image during content authentication. Tampered regions can be located correctly, and its original content can be recovered. The watermark information is secured by a secret key that randomises the watermark pixel positions. The single transform, correlator detector, and down-scaled processing spaces of the scheme offer low computational cost.*

**Keywords:** *Semi fragile watermark, self authentication, self recovery*

### 1.0 INTRODUCTION

Robust watermarks are suitable for copyright protection in a DRM scenario because it stays intact with the image persistently. On the other hand, fragile watermarks are good for tamper detection applications due to its ability in highlighting changes in images [1]. Recent development in the watermarking world witnesses the rise of semi fragile watermarks. As the name suggests, semi fragile watermarks resides in the grey area between the two extremes of robust and fragile watermarks. It possesses some properties of both robust and fragile watermarks. The need for semi fragile watermarks arises from the requirements of content authentication where the watermark must highlight malicious attacks while tolerating legitimate changes that do not alter the content severely. For example, a semi fragile watermark should give alerts under a cropping attack, and resist high quality image compression. Content authentication in this context is also named soft authentication. On the other hand, hard authentication is the validation of content that does not allow any modifications.

That means a single bit change in the test image will trigger an alarm that indicates the content is unauthentic.

## 2.0 CHALLENGES

Semi fragile watermarks had been studied in recent years and improvements had been made [2,3]. However, there are some challenges to be addressed. For instance, blind watermark detection and cropping resistant are hard to achieve in most watermarking schemes. In addition, the ability to reconstruct a cropped region in semi fragile watermarks can hardly be found in existing schemes [2,3,4,5]. Therefore, we set the goal to overcome these limitations. The requirements of our semi fragile watermarking are listed below:

1. It must allow mild image enhancements and compression that does not change the perceptual quality of the image. This resulted in the test images be classified as authentic.
2. It must alert users of malicious image modifications that affect visual quality of the image. This resulted in the test images be classified as unauthentic.
3. The watermark detection process must operate in a blind manner, i.e. without resort to a reference images. The watermark detection process here includes the detection, extraction, and decoding of watermark information. The reference image could be the cover image (a.k.a. host image) or the un-attacked stego image.
4. It must locate and highlight tampered regions correctly. This refers to tamper localization ability when tampering is detected.
5. An approximate content recovery must be made without the need of a reference image. This recovery could be necessary under cropping attack or region modifications that change its original content.
6. The watermark information must be secured so that adversaries cannot modify it without being detected.
7. The watermarking scheme must be balanced in terms of semi fragility, watermark imperceptibility, computational costs, and security.

## 3.0 SYSTEM DESIGN

Semi fragile watermarking schemes had been developed in spatial and transform domains. Spatial domain schemes usually exploit the statistical properties of the image pixels in detecting tampering and provide authentication. As such, their implementations are normally simple and fast. On the other hand, transform domain schemes offer robustness to compression by using the frequency information in the domain [6,7].

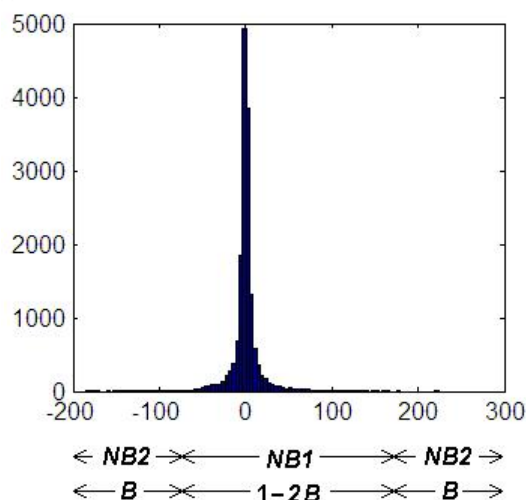
We chose to utilise spatio-temporal information in the wavelet domain in our semi fragile watermarking scheme. Discrete Wavelet Transform (DWT) offers both frequency information and spatial information. In contrast, Discrete Cosine Transform (DCT) and Discrete Fourier Transform (DFT) lack spatial information. Although DWT will certainly increase the computational load, we compensate it with simple

processing steps in its embedding and detection processes. For example, we use the low pass band and a down-scaled image in authentication and tamper localization instead of the commonly used block-base approach.

To fulfil the requirements listed in Section 2 above, we embed a down-scaled version of the image in the high pass bands of the wavelet coefficients. Embedding watermark in the high pass bands provides mild robustness to image compression. A higher level of robustness can be achieved by embedding the watermark in higher level subbands. However, this will degrade the visual appearance of the stego image. Therefore, we embed the watermark at the lowest level subbands.

The majority of semi fragile watermarking schemes employ block-based processing for authentication and tamper localization. For example, a mean value of  $8 \times 8$  pixel block can be embedded into a cover image. Later, it can be extracted from the stego image and compared with the computed mean value of a block at the same location to detect tampering. This approach certainly involves a large amount of computation. We reduce the computation load of the authentication and tamper localization by processing a down-scaled version of the cover image and the low pass band of the wavelet transform. The effect of such approach is the same as block-based approaches because each of the element in the down-scaled image or the low pass band corresponds to a certain block of pixels in the stego/test image. Due to the same reason, minor changes in the stego image would have minimal effect on the element values. Therefore, we can apply a simple correlator to detect tampering and localize it in the spatial domain.

Quantization was chosen as the embedding method in the wavelet domain due to its robustness [8]. Watermark is usually embedded in the high-pass subbands for better imperceptibility. We use the histogram of wavelet subbands to perform quantization for reduced computation. Besides, quantization also allows fine tuning by varying the number of quantization bins. A larger number of bins offer better imperceptibility at the cost of watermark extraction accuracy. This is because the bin size becomes smaller with a larger number of bins for a fixed range of coefficient values, and it means the changes made during watermark embedding will be smaller. At the same time, the watermark extraction accuracy would be degraded because the distinction between the bins becomes smaller. To find a balance point between these contradicting requirements, we propose a scheme with varying bin size. Analysing the histogram of high-pass subbands, it is noticed that most of the wavelet coefficients have near-zero values because it corresponded to flat regions in the image. Also, these coefficients occupy only a small range of the values in the histogram. We can use a small number of bins for these middle range coefficients because the changes made would be small. That means good imperceptibility and high watermark extraction accuracy. As for both ends of the histogram with large-value coefficients, we use a large number of bins to minimise changes in coefficient values for good imperceptibility. Although this would degrade the watermark extraction accuracy, the total effect is minimal because these coefficients only occupy a small fraction of the subband. Fig. 1 illustrates the histogram of a level-2 DWT horizontal subband. We use  $NB1$  bins for the  $(1-2B)$  part of the middle range coefficients. We also use  $NB2$  bins for the lower end and upper end coefficients. They are indicated as range  $B$  in Fig.1.



**Fig. 1: Histogram of a level-2 DWT horizontal subband**

The watermark bits are embedded in locations far away from its original position in order to combat cropping attack and enable content recovery. For example, watermark information of the lower right corner of the cover image can be embedded into the upper left corner of a wavelet subband. This way, a cropped area in the lower right corner of a stego image can be recovered by extracting watermark information from the un-affected upper left corner of the same image. In addition, the watermark embedding positions can be made random using a private key to offer security.

An overview of the watermarking scheme is depicted in Fig. 2. A watermark is generated by taking the down-scale version of the cover image. The cover image is transformed into the wavelet domain by DWT. Then, the higher order bits of the watermark are embedded into one wavelet subband, and the lower order bits are embedded into another wavelet subband. Following that, the stego image is obtained by an inverse transform from the wavelet domain into the spatial domain. To authenticate a test image which could have undergone changes, a DWT is performed and the watermark is extracted from the wavelet subbands. The watermark is then compared with a down-scale version of the test image. If the similarity between them exceeds a threshold value, then the test image is classified as authentic. Otherwise, tampered regions will be highlighted and content recovery is carried out using the watermark information extracted.

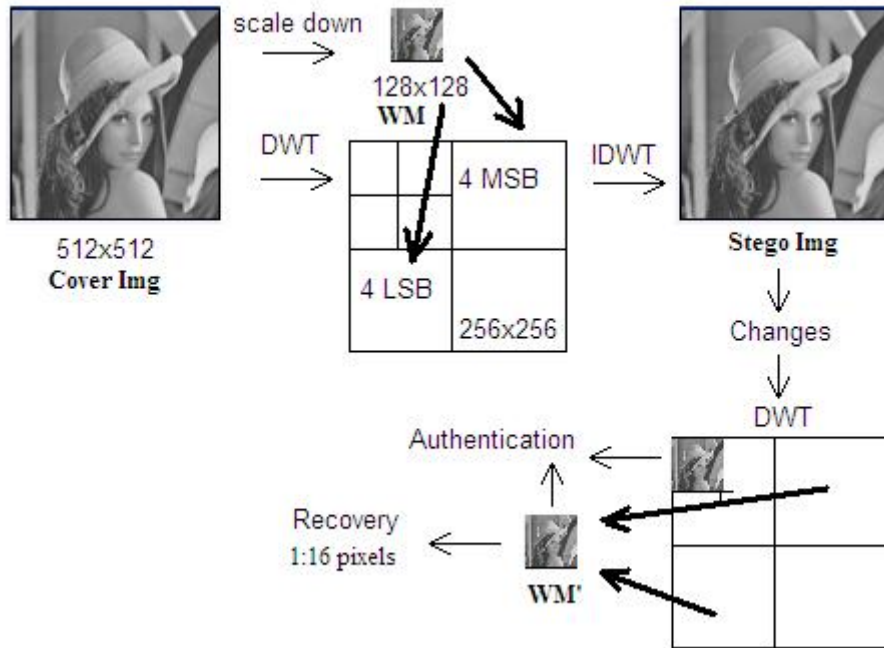


Fig. 2: Overview of the Semi Fragile Watermarking Scheme

### 3.1 Watermark Embedding

The watermark is generated by taking a down-scale version of the cover image in order to enable content authentication and self-recovery. Using an 8-bit gray scale watermark, we embed the 4 most significant bits (MSB) into the horizontal subband, and the 4 least significant bits (LSB) into the vertical subband. For enhanced security, a secret key can be used to map pixel positions from the watermark to the subbands. Then, an IDWT is performed using the embedded subbands to obtain a stego image.

The pseudo-code below describes details of the watermark embedding steps.

#### Embedding pseudo code

1. Initialize user defined parameters:
  - a. Let  $f(m,n)$  be the cover image
  - b. Let  $w(p,q)$  be the watermark for authentication
  - c. Let  $L \in \{1,2\}$  be the wavelet decomposition level for watermark embedding
  - d. Let  $N1 \in \mathbb{Z}^+$  be the count of quantization bins in the middle range of the wavelet subband histogram
  - e. Let  $N2 \in \mathbb{Z}^+$  be the count of quantization bins in both ends of the wavelet subband histogram
  - f. Let  $B=[0, 0.3]$  be the boundary fraction for both ends of the wavelet subband histogram
  - g. Let  $skey(p,q)$  be the secret key used to randomised pixel positions
2. Decompose  $f(m,n)$  using Haar filter for  $L$  levels. Let  $g_{k,l}(m,n)$  be the subbands  $k \in \{a,h,v,d\}$  at level  $l \in L$  of the wavelet coefficients.  $a$  refers to the approximate subband,  $h$  refers to the horizontal subband,  $v$  refers to the vertical subband, and  $d$  refers to the diagonal subband.

3. Take a down scale version of  $f(m,n)$  having the same size as  $g_{k,L+1}(m,n)$  as an initial watermark, then map its pixel positions using  $key(p,q)$  to produce the secure watermark  $w(p,q)$  for authentication and content recovery.
4. Construct a  $NI$ -bin quantization table for each  $h,v$ , and  $d$  subbands by taking the  $(1-2B)$  middle range of wavelet coefficients. Append  $N2$ -bin within  $B$  range to the quantization tables at both ends. A quantization function  $Q$  is used to map each wavelet coefficient to a binary value:

$$Q(f) = \begin{cases} 0 & \text{if } r\Delta \leq f < (r+1)\Delta \text{ for } r = 0, \pm 2, \pm 4, \dots \\ 1 & \text{if } r\Delta \leq f < (r+1)\Delta \text{ for } r = \pm 1, \pm 3, \pm 5, \dots \end{cases}$$

where  $\Delta$  is the quantization parameter:  $\Delta = [(f_{kmax} - f_{kmin}) \times (1-2B)] / NI$  for the middle range of wavelet coefficients; and  $\Delta = [(f_{kmax} - f_{kmin}) \times B] / N2$  for both the ending range of wavelet coefficients.

5. Embed 4 MSB of  $w(p,q)$  into the  $h$  subband, and 4 LSB of  $w(p,q)$  into the  $v$  subband. For example, we can choose to embed the 4 MSB of  $w(m,n)$  into  $g_{h,L}(m,n)$ ,  $g_{h,L}(m,n+1)$ ,  $g_{h,L}(m+1,n)$ , and  $g_{h,L}(m+1,n+1)$ . Based on the quantization table constructed, embedding is made as follows:

```

if  $Q(g_{k,l}(m,n)) = w(m,n)$ 
    set  $g_{k,l}(m,n) = (r+0.5)\Delta$ 
else
    if  $Q(g_{k,l}(m,n)) > (r+0.5)\Delta$ 
        set  $g_{k,l}(m,n) = (r + 1.5)\Delta$ 
    else
        set  $g_{k,l}(m,n) = (r - 1.5)\Delta$ 
    end
end
end

```

The embedding process ensures each wavelet coefficient maps to the correct bit value by assigning a value in the middle part of the quantization bin. It also ensures minimal changes when flipping a bit value by moving the current value to its next bin or previous bin. Note that  $\Delta$  should be small in order to minimize visual quality degradation, and this embedding process may shift the minimum and maximum values of wavelet coefficients by  $\Delta/2$  at both ends of the wavelet subband histogram.

6. Perform Inverse DWT using the embedded  $h$  and  $v$  subbands, and the original  $a$  and  $d$  subbands to obtain the stego image.

### 3.2 Watermark Detection

One must use the correct set of private information to detect the watermark correctly, i.e.  $L$ ,  $NI$ ,  $N2$ ,  $B$ , and  $key$ . The watermark detection begins with a DWT on the test image. Then, quantization table for each highpass subband is constructed. Following that, 4 bits of gray level information are extracted from the horizontal and vertical subbands respectively to form an 8-bit gray scale watermark. The extracted watermark is compared with the down-scaled version (alternative: use lowpass subband) of the test image to determine its authenticity. If the test image is not authentic, then tamper localization and approximate content recovery is carried out using a secret key and a threshold value.

Note that authentication is carried out in a blind detection manner because it does not require a reference image. However, due to its fragile nature, the watermark would be

destroyed in the case of a non-authentic test image. Therefore, tamper localization and content recovery would require a secret key or a reference image. To enable blind detection in this case, we use a down-scaled version of the cover image as the secret key.

The pseudo-code below describes details of the watermark detection steps.

**Detection pseudo code**

1. Initialize user defined parameters:
  - a. Let  $f'(m,n)$  be the test image. This is the stego image that could have undergone attacks
  - b. Let  $w(p,q)$  be the watermark for tamper localization and content recovery, this is the secret key that enables blind watermark detection
  - c. Let  $L \in \{1,2\}$  be the wavelet decomposition level for watermark embedding
  - d. Let  $N1 \in \mathbb{Z}^+$  be the count of quantization bins in the middle range of the wavelet subband histogram
  - e. Let  $N2 \in \mathbb{Z}^+$  be the count of quantization bins in both ends of the wavelet subband histogram
  - f. Let  $B=[0, 0.3]$  be the boundary fraction for both ends of the wavelet subband histogram
  - g. Let  $skey(p,q)$  be the secret key used to randomise pixel positions
  - h. Let  $t1 \in \langle \text{Real positive} \rangle$  be the threshold value for authentication
  - i. Let  $t2 \in \langle \text{Real positive} \rangle$  be the threshold value for tamper localization and content recovery
2. Decompose  $f'(m,n)$  using Haar filter for  $L$  levels. Let  $g_{k,l}(m,n)$  be the subbands  $k \in \{a,h,v,d\}$  at level  $l \in L$  of the wavelet coefficients, and  $a,h,v$ , and  $d$  are the same as defined in the watermark embedding pseudocode.
3. Construct a  $N1$ -bin quantization table for each  $h,v$ , and  $d$  subbands by taking the  $(1-2B)$  middle range of wavelet coefficients. Append  $N2$ -bin within  $B$  range to the quantization tables at both ends.
4. Extract the watermark  $w'(p,q)$  from the subbands at level  $L$ , taking 4 MSB from the  $h$  subband, and 4 LSB from the  $v$  subband. To do this, use the quantization function  $Q$  as in watermark embedding steps to map each wavelet coefficient to a binary value. Note that  $w'(p,q)$  has the same size as a subband at level  $L+1$ .
5. Reverse the mapping of pixel positions in  $w'(p,q)$  using information in  $skey(p,q)$  to produce the watermark  $w''(p,q)$  for authentication. The watermark  $w''(p,q)$  should appear as a down-scaled version of the cover image, with some error bits that may occur due to attacks. To reduce the error effects, we can perform a smoothing operation on  $w''(p,q)$ . This will also enhance the semi-fragile characteristic of the watermark for authentication by introducing some "fuzziness".
6. Compute the two-dimensional correlation coefficient,

$$corr2 = \frac{\sum_p \sum_q (w'' - \bar{w})(u - \bar{u})}{\sqrt{\left[ \sum_p \sum_q (w'' - \bar{w})^2 \right] \left[ \sum_p \sum_q (u - \bar{u})^2 \right]}}$$

where  $u(p,q)$  is the down-scaled version of  $f'(m,n)$ ,  $\bar{w}$  is the mean value of  $w$ ,

and  $\bar{u}$  is the mean value of  $u$ .

7. Use thresholding to determine the authenticity of the test image:
  - if  $corr2 > t1$   
The image is authentic
  - else  
The image is not authentic
  
8. To locate tampered regions in an unauthentic test image, a tampering matrix  $y(p,q)$  is computed and compared to the threshold value  $t2$ 

$$y(p,q) = u(p,q) - \underline{w}(p,q)$$
 where  $\underline{w}(p,q)$  is the down-scaled version of the cover image.
  - if  $|y(p,q)| > t2$   
 $y(p,q)$  is tampered
  - else  
 $y(p,q)$  is not tampered
  
9. To recover the contents of the tampered regions,  $y(p,q)$  is up-scaled to the size of the test image and the tampered regions are replaced by an up-scaled version of  $\underline{w}(p,q)$ . Since there are scaling operations involved here, the recovered content is an approximation instead of an accurate one. Although  $\underline{w}(p,q)$  is recommended here,  $w''(p,q)$  can be used for content recovery too.

#### 4.0 EXPERIMENTAL RESULTS ANALYSIS

This section describes the experiment settings and analyse the experimental results with regards to imperceptibility, semi fragile performance, tamper localisation, and content recovery.

##### 4.1 Experiment Settings

Four images with different characteristics are used in the experiment. *Baboon* has complex textures, *Lena* has clear boundaries between regions, *Pepper* has smooth surfaces, and *Fishing boat* has high contrast areas and tiny objects. The 512×512 pixel cover image  $f(m,n)$  is down-scaled to 64×64 pixel to form the watermark  $w(p,q)$ . The other settings are  $L = 2$ ,  $N1 = 22$ ,  $N2 = 400$ , and  $B = 0.25$ . For simplicity and ease of manual verification,  $key(p,q)$  is chosen as a circular shifted matrix in both horizontal and vertical directions. This shift at half of its size will produce a watermark with 4 quadrants having maximum distance from its original position, and can be powerful in fighting cropping attacks. An example of the watermark  $w(p,q)$  for *Lena* produced by  $key(p,q)$  is shown in Fig. 3. Note that a randomly permuted  $key(p,q)$  is preferred for high security system.





Fig. 3: (Left) The cover image *Lena*; (Right) its watermark to be embedded.

#### 4.2 Imperceptibility

The difference between a cover image and its stego image is minimal and does not reveal any information about the watermark because it appears as random noise. Fig. 4 illustrates an example. The PSNR of the stego images are 41.26 dB for *Lena*, 41.14 dB for *Baboon*, 40.15 dB for *Pepper*, and 40.11 dB for *Fishing boat*.



Fig. 4 (Left) The cover image; (Middle) The stego image; (Right) The magnified difference between the cover image and its stego image.

#### 4.3 Semi Fragile Performance

The parameters applied in watermark detection must be the same as its embedding procedures because this is a symmetric key watermark system. In addition, the threshold values  $t_1$  and  $t_2$  are determined through experiments. Higher threshold values increase its fragile nature and make it more sensitive to changes. For example,

$t1 = 0.86$  and  $t2 = 30.0$  for *Lena* shows optimal performance. Fig. 5 illustrates interim watermark detection results. Some error bits in watermark extraction can be seen when comparing the original watermark in Fig. 3 (Right) with the extracted one in Fig. 5 (Left). Smoothing operation was applied to reduce the error effects. The smoothing operation also provides “fuzziness” for its semi-fragility because exact comparison of content is not required. This is known as soft authentication in semi fragile watermarking. On the other hand, fragile watermarking scheme require hard authentication. Exact comparison of content is performed in a hard authentication.



Fig. 5 (Left) The 8-bit gray scale watermark  $w'(p,q)$  extracted with some error pixels; (Middle) The error-reduced  $w''(p,q)$  produced from remapping  $w'(p,q)$  and smoothing; (Right) The down-scaled version of the test image for authentication.

Table 1 lists suitable authentication threshold values  $t1$  for each test image after examining its corresponding correlation  $corr2$  values. All of the test images were watermarked using the parameter values mentioned in Section 4.1 above. Local shift attack was performed by shifting the region (130:220,115:125) five columns to its right, and shifting the region (382:392,260:340) two rows upwards. Noise attacks involved adding “salt and pepper” noise with varying density. JPEG compression attacks used quality factors of 90, 80, and 70. Shift attacks involved circular shift with varying row and column. Rotation attacks are rotation at 1,2, and 4 degrees with auto-cropping. Cropping attacks cropped off a rectangular region of the stego images by setting its pixels to zero value. Mean filtering attacks have kernel size ranging from  $2 \times 2$  to  $5 \times 5$ . Sample images of these attacks are included in the Appendix. To allow high quality modifications that do not affect visual quality of the images, the threshold value for each image was selected so that test images underwent local shift, low level of noise insertion, and high quality JPEG compression are classified as authentic. It is observed that *Baboon* has the lowest threshold at 0.70 whereas *Pepper* has the highest threshold at 0.88. This can be explained by the complexity of the image texture. Overall, *Baboon* has the most complex texture and it caused the lowest correlation value  $corr2$  in authentication because the extracted watermark is severely distorted. Adversely, *Pepper* has smooth textures, thus its watermark has the highest correlation value. The use of correlation-based thresholding is inherently weak to shifting attacks. For example, circular shift of one row does not affect the visual quality of the image but the result will be classified as non-authentic.

**Table 1: Semi fragile authentication under various attacks and threshold selection**

Attack	Attack level	<i>corr2 value</i>			
		<i>Lena</i>	<i>Baboon</i>	<i>Pepper</i>	<i>Fishing boat</i>
No attack		0.88	0.72	0.88	0.83
Local shift		0.87	0.72	0.88	0.83
Histogram equalisation		0.29	0.22	-0.21	-0.14
Noise	0.0005	0.87	0.72	0.88	0.83
	0.001	0.87	0.72	0.88	0.82
	0.005	0.84	0.67	0.85	0.78
JPEG compression	90	0.87	0.71	0.88	0.83
	80	0.68	0.59	0.87	0.76
	70	0.68	0.49	0.82	0.15
Shifting	[1 0]	-0.10	-0.01	0.02	0.00
	[0 2]	0.20	-0.01	-0.18	0.11
	[3 0]	0.06	-0.02	0.09	-0.08
	[2 2]	-0.05	0.02	-0.01	0.05
Rotation	1° and crop	0.02	0.02	-0.05	0.01
	2° and crop	0.06	0.08	-0.13	0.03
	4° and crop	0.01	-0.03	0.08	0.08
Cropping	(1:50,460:512)	0.84	0.70	0.85	0.78
	(1:90,420:512)	0.79	0.65	0.79	0.71
Mean filtering	2x2	-0.46	-0.02	0.55	0.37
	3x3	-0.34	-0.13	0.54	-0.35
	4x4	0.12	-0.03	-0.06	-0.07
	5x5	0.02	0.05	0.07	-0.07
<b>Threshold <i>t1</i></b>		<b>0.86</b>	<b>0.70</b>	<b>0.88</b>	<b>0.81</b>

Besides correlation, PSNR value can also be used in authentication because it is based on the same principle of measuring the likelihood between two images. Therefore, the PSNR value calculated using the extracted watermark and the down-scaled version of the test image can replace the correlation value in image authentication. Table 2 lists the PSNR value for each image under various attacks. Based on those results, suitable threshold values  $t1$  for each image are also suggested.

**Table 2: Alternative semi fragile authentication and threshold selection**

Attack	Attack level	PSNR value			
		<i>Lena</i>	<i>Baboon</i>	<i>Pepper</i>	<i>Fishing boat</i>
No attack		20.98	18.76	20.12	19.78
Local shift		20.69	18.73	20.04	19.74
Histogram equalisation		10.08	10.77	9.40	9.41
Noise	0.0005	20.85	18.71	19.93	19.65
	0.001	20.70	18.65	20.02	19.44
	0.005	19.88	18.06	19.08	18.66
JPEG compression	90	20.84	18.48	20.13	19.64
	80	17.07	17.29	19.75	18.46
	70	17.25	16.49	18.43	14.02
Shifting	[1 0]	11.96	13.83	12.19	12.46
	[0 2]	12.52	14.13	11.64	13.73
	[3 0]	11.39	13.86	11.13	12.36
	[2 2]	12.59	14.16	12.53	13.50
Rotation	1° and crop	13.09	14.09	11.24	13.16
	2° and crop	13.54	14.22	11.83	13.21
	4° and crop	13.22	13.53	11.51	13.15
Cropping	(1:50,460:512)	19.81	18.37	19.23	18.76
	(1:90,420:512)	18.63	17.60	17.88	17.29
Mean filtering	2x2	11.35	14.47	14.80	15.02
	3x3	11.53	14.11	14.88	12.02
	4x4	11.71	15.12	12.10	13.29
	5x5	10.51	15.00	12.12	12.58
<b>Threshold <math>t1</math></b>		<b>20.00</b>	<b>18.40</b>	<b>19.90</b>	<b>19.00</b>

#### 4.4 Tamper Localisation

Tamper localisation is performed if a test image is not authentic. Tampered regions are detected by comparing the extracted watermark with the down-scaled version of the test image. Instead of up-scaling the watermark to the size of the test image for authentication, we down-scale the test image to the size of the watermark to reduce computation. Fig. 6 illustrates an example of tamper localisation. The unaltered stego image is in the top left corner. Tampering was done by copying the flower knot near the edge of the hat and pasting its magnified version onto the centre of the hat. The result is shown in the top right corner. Tamper localisation correctly highlighted the tampered region as depicted in the bottom left corner. However, a small area of the tampered region was not classified as tampered region due to the selected threshold values of  $t1$  and  $t2$ . This demonstrates the semi-fragile nature of the watermarking scheme. In order to achieve high level of fragility, a high value of threshold can be chosen for authentication.



Fig. 6 (Top left) The unaltered stego image; (Top right) The test image with tampered hat; (Bottom left) Detected tampered region in black colour; (Bottom right) Recovered image.

#### 4.5 Content Recovery

Although correlation-based authentication is not new, semi fragile watermarking systems that offer content recovery under cropping attack is very rare. This watermarking system provides tamper localization and approximate content recovery. The tampered regions can be identified correctly, and the approximately recovered contents give the user an idea of the image regions altered. Such information can be useful for human judgement in determining the severity of tampering. The bottom right corner of Fig. 6 depicts the approximately recovered content of the tampered region. The recovery is done using the extracted watermark information after localising tampered region. Comparing the recovered image in the bottom right corner with the original stego image in the top left corner of Fig. 6, the recovered content was nearly identical. However, due to the limited amount of watermark information embedded, the recovery cannot provide detailed information such as complex textures and crisp edges. For example, Fig. 7 illustrates a stego image with its top left corner cropped off, and the approximately recovered content. The self authentication and self

recovery capabilities of this watermarking scheme made it practical in real life scenario where a reference image may not be available.



Fig. 7 (Left) Top-left corner cropping on the stego image (Right) Approximately recovered content without edge details.

One major weakness of semi-fragile watermark in content recovery is its fragile nature. If a large region of the test image is cropped off, then the watermark information is lost and content recovery is impossible. Similarly, if a large region of the test image undergone severe distortion, then the watermark information is lost. Therefore, content recovery is impossible when a large region of the test image is distorted. To have good performance in content recovery, a robust watermark is needed.

#### 4.6 False Positive Condition

To evaluate the watermarking scheme under false positive condition, all of the 4 images were not embedded with any watermark and sent to watermark detection step. Each of the images was tested 10 times in watermark detection. The results appeared as random noise. These indicated that the watermarking scheme works correctly.

#### 4.7 Watermark Security

Security of the watermark is achieved by randomising watermark pixel positions using the secret key  $key(p,q)$ . This is necessary to deter malicious attacks when the watermarking algorithm is made public. For a watermark of  $64 \times 64$  pixels and 256 gray scales, there are  $(64 \times 64 \times 256)! = (2^{20})!$  possible combinations. If we simplify the problem with binary watermark, there is  $(64 \times 64 \times 2)! = (2^{13})!$  possible combinations. Together with other watermarking parameters such as the quantization bin count, the boundary fraction, and the threshold value, an adversary would have to try a huge number of combinations in order to break the system.

## 5.0 CONCLUSIONS

Semi fragile watermark is suitable for content authentication where legitimate modifications are allowed and malicious attacks are highlighted. Based on the limitations of current watermarking schemes, system objectives were listed to address the challenges. We developed and evaluated a semi fragile watermarking scheme that offers self authentication and self recovery.

Our scheme provides content authentication by allowing high quality JPEG compression, minor local distortion, and minimal noise insertion. Other changes such as histogram equalisation, cropping, rotation, and mean filtering are classified as malicious attacks because it affects the visual quality of the image. The scheme is practical because it does not require a reference image during content authentication. Tampered regions can be located correctly, and its original content can be recovered. The approximately recovered content could give the user an idea of the original image appearance. The watermark information is secured by a secret key that randomises the watermark pixel positions. The single transform, correlator detector, and down-scaled processing spaces of the scheme offer low computational costs.

The watermarking scheme is inherently unable to recover image content if exposed to severe attacks such as a major cropping. This vulnerability must be overcome by a robust watermark. In addition, due to the adoption of correlator detector, the watermarking scheme cannot classify minor shift as legitimate modification. One way to overcome this is to divide the image into blocks and watermark each block separately. However, this will increase the computational costs. To balance imperceptibility and semi fragility, watermark is embedded in the 2<sup>nd</sup> level of wavelet subbands. This resulted in a downgrade of accuracy in tamper localisation.

Overall, the watermarking scheme achieved its objectives in providing self authentication and self recovery in a semi fragile manner. A hybrid system that combines robust and semi fragile watermarks is recommended to overcome the weaknesses identified.

## REFERENCES

- [1] Cox, I, M.L.Miller, and J.A.Bloom, *Digital Watermarking*, 2001, San Francisco, California, USA, Morgan Kaufmann Publishers Inc., 539 pages.
- [2] Lin, C.Y. and S.F. Chang, "SARI:Self-Authentication-and-Recovery Image Watermarking System", *ACM Multimedia*, 2001, Ottawa, Canada: ACM Press. pp.628-629.
- [3] Rey, C. and J.L.Dugelay, "A Survey of Watermarking Algorithms for Image Authentication". *EURASIP Journal on Applied Signal Processing*, 2002. 2002(6):p.613-621.

- [4] Fridrich, J. and M.Goljan, "Images with Self-correcting Capabilities," in *International Conference on Image Processing 1999 (ICIP'99)*, 1999, Kobe, Japan, IEEE, pp.792-796.
- [5] Rey, C. and J.L.Dugelay, "Blind Detection of Malicious Alteration on Still Images using Robust Watermarks," in *IEE Seminar on Secure Images and Image Authentication*, (Ref.No. 2000/309), 2000, London, UK, pp.7/1-7/6.
- [6] Kundur, D. and D. Hatzinakos, "Digital Watermarking using Multiresolution Wavelet Decomposition," in *International Conference on Acoustic, Speech and Signal Processing (ICASP) 1998*, Seattle, Washington, USA, IEEE, pp.2969-2972.
- [7] Pereira, S., S. Voloshynovskiy, and T.Pun, "Optimized Wavelet Domain Watermark Embedding Strategy using Linear Programming," in *SPIE AeroSense 2000*, 2000, Orlando, Florida, USA, SPIE.
- [8] Chen, B. and G.W.Wornell, "Quantization Index Modulation: a Class of Provably Good Methods for Digital Watermarking and Information Embedding", *IEEE Transactions on Information Theory*, 2001, 47(4), p.1423-1443.

## **BIOGRAPHY**

Woo Chaw Seng is a senior lecturer at the Faculty of Computer Science and Information Technology, University of Malaya. His research interests include image processing and mobile applications.





Jiang Du is a staff at the Queensland University of Technology.






Binh Pham is an adjunct professor at the Queensland University of Technology.



**APPENDIX**

Sample images of various attacks.

Attack	Attack level	Authentic	Attacked image
No attack		Yes	
Local shift		Yes	
Histogram equalisation		No	
Noise	0.0005	Yes	

Noise	0.001	Yes	
	0.005	No	
JPEG compression	90	Yes	
JPEG compression	80	No	
JPEG compression	70	No	

Shifting [1 0] No



Shifting [0 2] No



Shifting [3 0] No



Shifting [2 2] No



Rotation 1° and crop No



Rotation 2° and crop No



Rotation 4° and crop No



Cropping (1:50,460:512) No



Cropping (1:90,420:512) No



Mean filtering 2x2 No



Mean filtering      3x3      No



Mean filtering      4x4      No



Mean filtering      5x5      No

