

Image Authentication and Self-Recovery Scheme Based on The Rehashing Model

Wan-Li Lyu^{1,2}, Chin-Chen Chang^{2,3}, Feng Wang⁴

¹Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education,
School of Computer Science and Technology,
Anhui University, Hefei 230039, China
wanly_lv@163.com

²Department of Information Engineering and Computer Science,
Feng Chia University,
100 Wenhwa Rd., Seatwen, Taichung 40724, Taiwan, ROC
alan3c@gmail.com

³Department of Computer Science and Information Engineering
Asia University, Taichung 41354, Taiwan, ROC
alan3c@gmail.com

⁴Department of Mathematics and Physics, Fujian University of Technology,
Fuzhou, Fujian, 350108, China
w.h.feng@163.com

Received October, 2014; revised December, 2015

ABSTRACT. *An effective image tamper localization and self-recovery scheme that uses the rehashing model is proposed in this paper. The aim of the proposed scheme was to reduce the failure rate in detecting image tamper areas and achieve high rates of successful detection. To achieve this aim, the proposed scheme uses a set of hash functions which are called the rehashing model, gained an optimal decreasing collision method to detect tamper areas. With an improved and accurate tamper detection rate, the tampered image can be recovered more accurately. The quality of the proposed scheme was tested with gray and color images. The experimental results indicated that the proposed scheme has good performance in locating tamper locations and allows image recovery with an acceptable visual quality up to 50% of the image content tampered.*

Keywords: Data hiding; Image authentication; Image recovery; Rehashing

1. Introduction. When we get an image from the Internet, we want to confirm the integrity of the image and regain the original image even if the visual quality has been diminished by tampering. In the past decades, several self-recovery watermarking schemes have been proposed for this purpose [1–12]. These watermarking schemes are focused on developing self-embedding methods, i.e., how to extract the primary features of the original image with as few bits as possible and how to embed the features into the original image with minimal distortion. The common solution is partitioning the original image into blocks. Block-based watermarking authentication methods use a few bits to represent the features of a block as the watermark to be embedded into another block later. Thus, the average intensity of a block, important quantized DCT coefficients or Integer Wavelet Transform coefficients can verify whether tampering has occurred and where it is located in the image. These features will continuously influence the subsequent work of

authentication and recovery. If ideal features of blocks can be obtained, can we localize the tamper successfully? For example, a self-recovery watermarking scheme can embed features of one block i into another block j , but when block i is tampered, the features extracted from block j would not be equal to the features of tampered block i , which complicates the tamper localization. Lin et al [8] proposed that the payload of the watermark consists of authentication data embedded into the block itself and recovery data embedded into another block. Similar approaches have been proposed using chaotic or random numbers [9–11] to help generate the authentication data. Some block-based watermarking authentication schemes concatenate a watermark unit with a random number [10, 12], which also is sent to the receiver, with the aim of locating tamper regions more accurately. Only the untouched block has the correct random values, so the tampered block can be located. Because the embedding capacity of watermarking algorithms is limited and the payloads also include essential recovery data, the length of random numbers should be very short. Many blocks have the same random number, making it impossible to locate many of the tamper units. In fact, the design of the random number is important to avoid unsuccessful authentication results.

Several hashbased image authentication methods [13–18] have extracted features from an image and projected these features with hash functions to gain hash value to authenticate the integrity of the image. The hash value of the image is organized with a tag and then appended to the original image I . It is a challenge to find a valid way from the image pixel space project to hash feature space because there can be countless arrangements of the image's pixel space. The tag's storage and transmission also are puzzling problems.

In fact, hashing is a key-to-address transformation technique in which the key space can be mapped into the address space. It is considered to be an effective means of organizing and retrieving data in practical applications, such as database management, compiler construction, and many other applications [19]. When more than two different keys have the same hashing value, collisions occur. If a hash function has the feature that from the set of keys in the key space to the address space is one-to-one, the hash function is much easier to use because the key collision problem can be avoided. Sprugnoli [19] called the one-to-one hash function the 'perfect hashing function.' Du et al. [20] proposed a rehashing model to design a perfect hash function with an indicator table called the hash indicator table (HIT). With the help of the HIT, several random hash functions can be organized to construct the desired perfect hash function. The rehashing model uses a number of hash functions, i.e., h_1, h_2, \dots, h_s , to reduce collisions. In our proposed scheme, this feature of the rehashing model can decrease the incidents of incorrect location of the tamper blocks.

The proposed scheme aims at locating tamper regions more accurately, and it innovatively utilizes the rehashing method for image authentication to overcome easy collisions of the random numbers to enhance the ability to resist attacks. The proposed scheme also can be used to authenticate integrity and recover tempered color images. To verify that using the rehashing model can improve the performance of tamper localization, the proposed scheme simply uses average intensity as a feature and the least significant bit (LSB) plane for embedding the watermark.

The rest of the paper is organized as follows. Section 2 briefly introduces and discusses Du et al's rehashing technology [20]. In Section 3, our image authentication and self-recovery scheme based on rehashing is described in detail. The experimental results are reported in Section 4. Our conclusions are presented in Section 5.

2. Du et al's Rehashing Method [20]. Assume that the key space has n distinct keys, K_1, K_2, \dots, K_n and assume that the address space has m entries. The use of a single hash

function, h_i which is selected randomly from $F_{n \times m}$, which is the set of the map from the key space to the address space, would result in many collisions in the address space. Let $P_i(m, n)$ denote the probability that the hash function has $i(0 \leq i \leq \min(m, n))$ entries in the address space with only one key hashed to them. The $P_i(m, n)$ can be computed as shown in Equation (1):

$$P_i(m, n) = \frac{e_i(m, n)}{m^n}, \text{ where } e_i(m, n) = n! \binom{m}{i} \sum_{r=0}^{n-i} (-1)^r \binom{m-i}{r} \frac{(m-r-i)^{n-r-i}}{(n-r-i)!} \quad (1)$$

The expected values of i for $P_i(m, n)$ are 3.8742, 7.5471, and 11.2240, and the probabilities are 167%, 0.020501% and 0.00030913% when $i \geq 0.8n$ and $m = n = 10, 20, 30$ respectively. The expectation of i as $P_i(m, n)$ approaches $\frac{n}{e}$ is very large when $m = n$.

However, the use of a number of hash functions, h_1, h_2, \dots, h_s , which are called the rehashing model, can eliminate many collisions. Fig. 1 shows the rehashing model using seven hash functions from key space with n entries to the address space also with n entries.

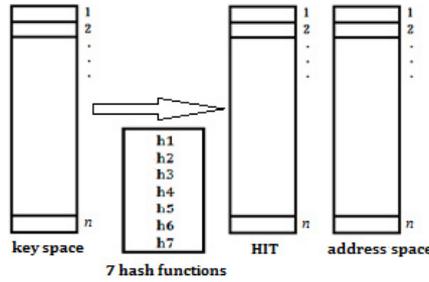


FIGURE 1. Rehashing model with seven hash functions from key space to address space

Fig.1 shows that, for a key in key space, the sequence numbers of the selected hash functions are stored in the HIT and the corresponding address is stored in the address space. In this case, the key space has n entries and the HIT also has n entries

Let $P_i^s(m, n)$ denote the probability that the address space has i ($0 \leq i \leq \min(m, n)$) entries with only one key hashed to them. The $P_i^s(m, n)$ can be computed with Equation (2):

$$P_i^s(m, n) = \sum_{r=0}^i P_r^{s-1}(m, n) \cdot Q_{i-r}(m, n, r) \quad (2)$$

where: $Q_i(m, n, j) = \binom{j}{i}^{n-j} \sum_{r=0}^{n-j} \binom{n-j}{r} \frac{e_i(r, m-j)}{j^r}$, and $P_i^1(m, n) = P_i(m, n)$

The expected values of i for $P_i^7(m, n)$ are 8.8000, 17.4644, and 26.0698, and the probabilities are 9641%, 9717% and 9784% when $i \geq 0.8n$ and $m = n = 10, 20, 30$ respectively. Our proposed scheme, which uses the rehashing model rather than a single hashing function eliminates numerous collisions, is suitable for calculating the address and can be used for processing the authentication messages.

3. Proposed Image Authentication and Recovery Scheme. The proposed scheme takes two pixels per unit, and one pixel of the unit embeds HIT information of itself, and the other pixel embeds recovery data of another unit. Fig. 2 illustrates a brief arrangement of three units and data hiding directions.

Fig. 2 shows that each pixel pair consists of a left pixel and a right pixel, e.g., unit A consists of A_L and A_R and the features of the pixels of unit A are embedded into the

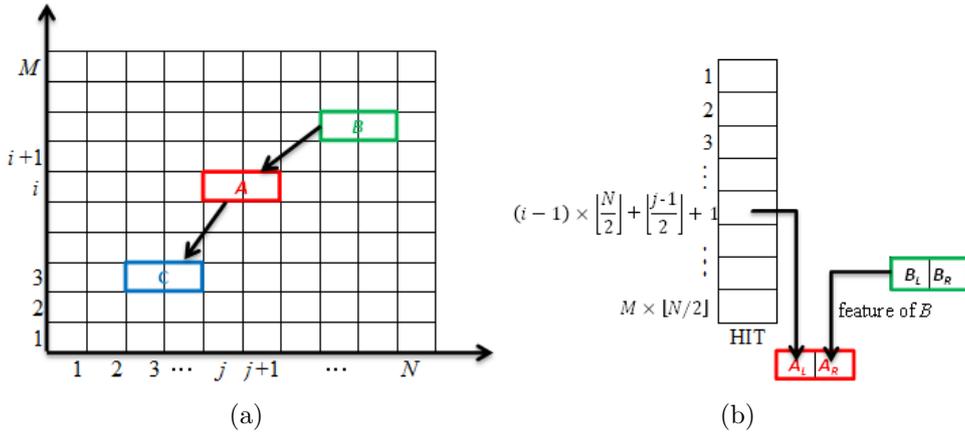


FIGURE 2. Illustration of brief unit arrangement of image and data hiding directions: (a) a pair of pixels per unit; (b) HIT and feature-hiding directions

LSBs of C_R , and the embedded content of A_R is unit B 's features. The LSBs of $A_L B_L$, and C_L are their corresponding contents in HIT.

The proposed image authentication and recovery scheme is described into five phases, i.e., the design of seven random hash functions and HIT construction, hiding HIT and feature secret bits, extracting HIT and feature secret bits, tamper detection, and image recovery.

3.1. Design seven random hash functions and construct HIT. Because the proposed scheme takes two pixels per unit, assume that the size of image I is $M \times N$ where $N \bmod 2 = 0$, as Fig. 2(a) shows. Use S as the seed for a random number generator to get seven random numbers, S_1, S_2, \dots, S_7 , and then use them as the random number generators to construct seven hash functions h_1, h_2, \dots, h_7 , each of which has the size of $M \times \frac{N}{2}$, and the range of the values of the hash functions h_1, h_2, \dots, h_7 , is $[1, M \times \frac{N}{2}]$. That is $1 \leq h_i(x) \leq M \times \frac{N}{2}$, where $1 \leq i \leq 7$ and $1 \leq x \leq M \times \frac{N}{2}$.

For an image, many different pixels have the same pixel value. Having the same pixel values can lead to having the same hashing values, so the pixels cannot be distinguished from each other and they are not suitable for use as hash keys. However, the locations of all of the pixels in an image are distinctive, so the proposed scheme uses the locations of pixels as the keys of hash functions. Using the rehashing method, the proposed scheme designs the HIT and hash address table (HAT) and inverse hash address table (IHAT) of image I . The procedure of constructing the HIT HAT and IHAT of image I is described as Algorithm 1.

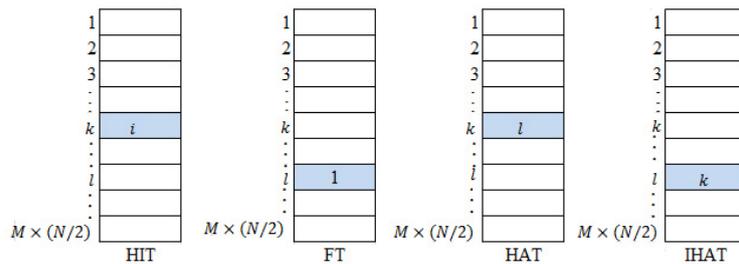


FIGURE 3. Arrangement of HIT, FT, HAT, and IHAT

Algorithm 1 Construct HIT, HAT, and IHAT suitable for image I

Input: A random number seed S and the size of image $I : M \times N$

Output: HIT, HAT, IHAT suitable for image I and a flag table FT

```

1: HIT[1:M ×  $\frac{N}{2}$ ] = 0
2: HAT[1:M ×  $\frac{N}{2}$ ] = 0
3: IHAT[1:M ×  $\frac{N}{2}$ ] = 0
4: FT[1:M ×  $\frac{N}{2}$ ] = 0
5:  $i = 1$ 
6:  $k = 1$ 
7: while  $k \leq M \times \frac{N}{2}$  do
8:   if  $i \leq 7$  and FT( $h_i(k)$ ) = 0 then
9:     FT( $h_i(k)$ ) = 1
10:    HIT( $k$ ) =  $i$ 
11:    HAT( $k$ ) =  $h_1(k)$ 
12:     $k = k + 1$ 
13:   else if FT( $k_i(k)$ )  $\neq 0$  then
14:      $i = i + 1$ 
15:   else  $i = 1$ 
16:      $k = k + 1$ 
17:   end if
18: end if
19: end while
20: for  $k = 1 : M \times \frac{N}{2}$  do
21:   if HIT( $k$ ) = 0 and FT( $k$ ) = 0 then
22:     HAT( $k$ ) =  $k$ 
23:     FT( $k$ ) = 1
24:   end if
25: end for
26: for  $k = 1 : M \times \frac{N}{2}$  do
27:   IHAT(HAT( $k$ )) =  $k$ 
28: end for
29: output HIT, HAT, IHAT and FT

```

Algorithm 1 uses a flag table (FT) to indicate whether the address table unit is occupied to speed up the execution time. Fig. 3 shows an example of the arrangement of HIT, FT HAT and IHAT. Assume that the hash function h_i , is suitable for the k^{th} unit of image I and $h_i(k) = l$, then $HIT(k) = i$ $FT(k) = 1$ $HAT(k) = l$ and $IHAT(l) = k$

In fact, only HIT is absolutely necessary among HIT, HAT and IHAT because with HIT and rehashing functions, we can compute HAT and IHAT. The use of HAT and IHAT can decrease the operating time in hiding the authentication message and in the extracting and image recovery phases.

3.2. Construct Authentication Image I' . The proposed scheme can authenticate the integrity of a gray image or a color image. For a color image, we must convert the color image from RGB color space to YUV color space. In the image recovery phase, we must convert the color image from YUV color space to RGB color space.

The proposed scheme uses average intensity as a feature and the least significant bit (LSB) plane for watermark embedding, which are described as Algorithm 2

Before construct authentication image, the scheme use Algorithm1 to construct HIT HAT and IHAT of image I . Segment image I into 1×2 blocks, $B_1, B_2, \dots, B_{M \times \frac{N}{2}}$

Algorithm 2 Construct authentication image

Input: $B_1, B_2, \dots, B_{M \times \frac{N}{2}}$

Output: The authentication image block $B'_1, B'_2, \dots, B'_{M \times \frac{N}{2}}$

1: **for** $k = 1 : M \times \frac{N}{2}$ **do**

2:

$$(B_k)'_L = (B_k)_L - (B_k)_L \bmod 8 + \text{HIT}(k) \quad (3)$$

3:

$$(B_k)'_R = (B_k)_R - (B_k)_R \bmod 8 + \text{msb}(B_{\text{IHAT}(k)}) \quad (4)$$

4: **end for**

5: output $B'_1, B'_2, \dots, B'_{M \times \frac{N}{2}}$

In Equation (3), the HIT has already been computed with Algorithm 1

For a block B_k , the Equation (4) compute features of block B_k with $\text{msb}(B_{\text{IHAT}(k)})$. the bit plane of $\text{mean}(B_k)$ is the mean value of B_k . The Algorithm 2 uses three MSBs as features of block B_k as Fig. 4 shows.

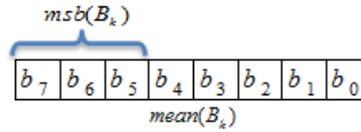


FIGURE 4. Plane of $\text{mean}(B_k)$ and the feature value $\text{msb}(B_k)$

Reconstruct blocks $B'_1, B'_2, \dots, B'_{M \times \frac{N}{2}}$ and the authentication image I' can be obtained.

For an RGB image I the proposed scheme first converts the image I from RGB color space to YUV color space and gains I_Y, I_U , and I_V . Using Algorithm 2, input I_Y, I_U and I_V separately, then obtain I'_Y, I'_U and I'_V . With I'_Y, I'_U and I'_V , we can convert the color image from YUV color space to RGB color space to get a color authentication image.

3.3. Procedure of image tamper detection and image recovery. The receiver receives the authentication image I' , which was generated with Algorithm 2 and the random number S which was used in Algorithm 1; then he or she can verify whether tampers have occurred in the content of the image.

Before verify the reality of the potential tampered gray image I' , the proposed scheme uses Algorithm 1 and the random number S to construct HIT suit for image I' . Segment the image I' into 1×2 blocks $B'_1, B'_2, \dots, B'_{M \times \frac{N}{2}}$. A tamper location map (TLM) that has the size of $M \times \frac{N}{2}$ is used in Algorithm 3.

The elements of TLM are $\text{TLM}(1), \text{TLM}(2), \dots, \text{TLM}(M \times \frac{N}{2})$. Let $\frac{N}{2}$ elements a line, then construct the TLM with M rows Use Equation (5) to draw the tamper position location image I_{Tamper}

$$\begin{cases} I_{\text{Tamper}}(\frac{k-1}{N/2} + 1, 2 \times ((k-1) \bmod (N/2) + 1) - 1) = \text{TLM}(k) \\ I_{\text{Tamper}}(\frac{k-1}{N/2} + 1, 2 \times ((k-1) \bmod (N/2) + 1)) = \text{TLM}(k) \end{cases}, \quad k = 1, 2, \dots, M \times \frac{N}{2}. \quad (5)$$

Algorithm 3 Verify the reality of image I' **Input:** The image blocks, $B'_1, B'_2, \dots, B'_{M \times \frac{N}{2}}$ **Output:** The elements of TLM are $\text{TLM}(1), \text{TLM}(2), \dots, \text{TLM}(M \times \frac{N}{2})$

```

1: for  $k = 1 : M \times \frac{N}{2}$  do
2:   if  $(B'_k)_L - \lfloor (B'_k)_L / 8 \rfloor == \text{HIT}(k)$  then
3:      $\text{TLM}(k) = 1$ 
4:   else
5:      $\text{TLM}(k) = 0$ 
6:   end if
7: end for
8: output TLM

```

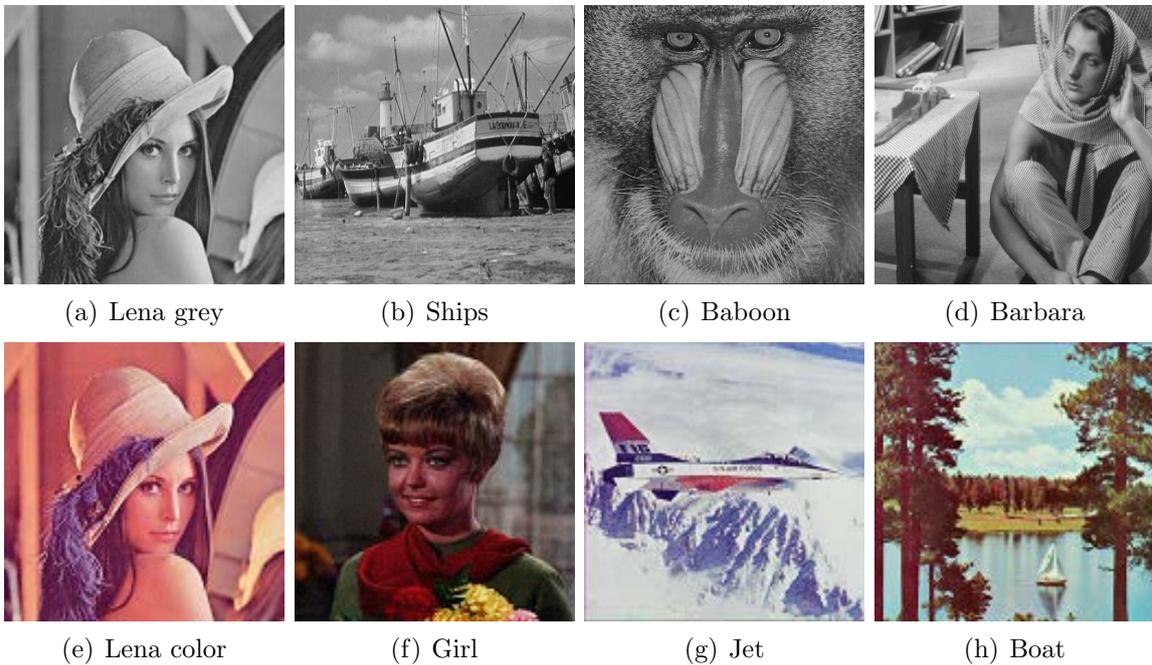


FIGURE 5. Original test images

For a potential tampered RGB image I' before executing Algorithm 3, the proposed scheme first converts the image I' from RGB color space to YUV color space and gains I'_Y , I'_U and I'_V . Using Algorithm 3 input $I'_Y, I'_U,$, and I'_V separately, then obtain $I_{YTamper}$, $I_{UTamper}$ and $I_{VTamper}$. Use Equation (6) to get the tamper position location image I_{Tamper}

$$I_{Tamper}(i, j) = I_{YTamper}(i, j) \oplus I_{UTamper}(i, j) \oplus I_{VTamper}(i, j) \quad (6)$$

where: $1 \leq i \leq M$, $1 \leq j \leq N$ and \oplus is an OR operator. If more than one out of three planes is tampered, $I_{Tamper}(i, j)$ should be marked as “tampered”.

Before recover the potential tampered gray image I' , the scheme uses Algorithm 1 and the random number S to construct IHAT suit for image I' . Segment the image I' into 1×2 blocks, $B'_1, B'_2, \dots, B'_{M \times \frac{N}{2}}$, and initialize a Recovery Map (RM) that has the size of $M \times \frac{N}{2}$ and the element values are $mean(B'_k)$

The elements of RM are $\text{RM}(1), \text{RM}(2), \dots, \text{RM}(M \times \frac{N}{2})$. Each row vector consists of $\frac{N}{2}$ elements, then construct the RM with M rows. Use Equation (7) to draw the recovery

Algorithm 4 Recover the potential tampered gray image I'

Input: The potential tampered gray image I' , which had the size of $M \times N$, the random number S , which was used in Algorithm 1, and TLM, which was used in Algorithm 3

Output: Recovery image $I_{recovery}$

```

1: for  $k = 1 : M \times \frac{N}{2}$  do
2:   if  $TLM(k) == 0$  then
3:      $RM(k) = (B_k)'_R \bmod 8 \times 32$ 
4:   end if
5: end for
6: output RM

```

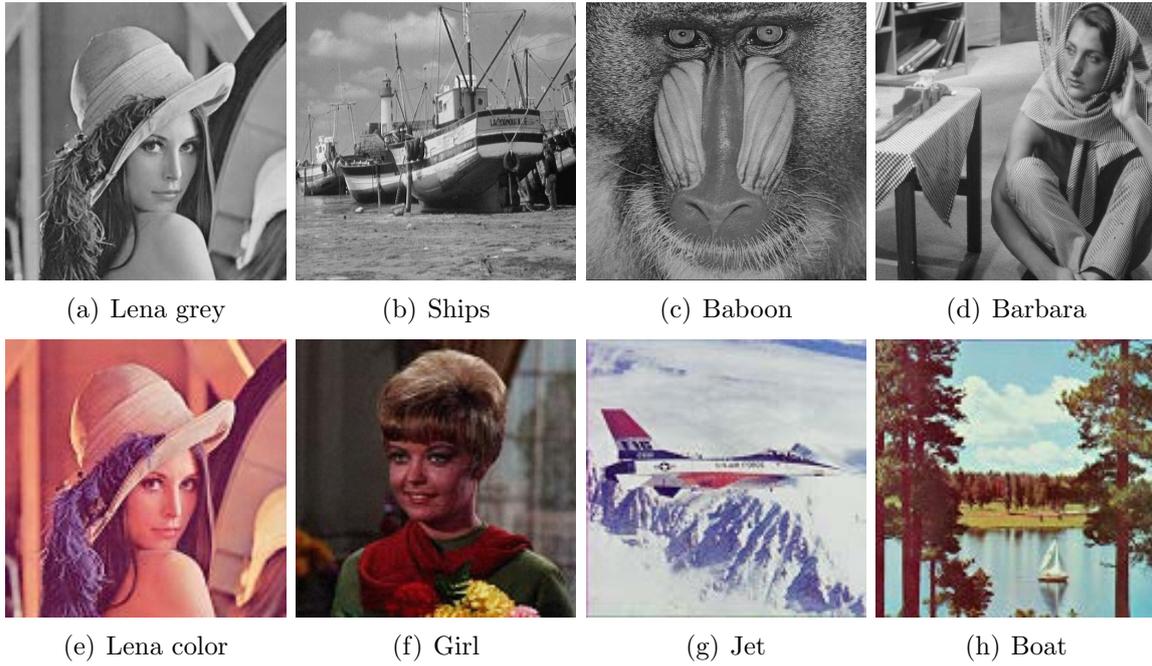


FIGURE 6. Images with embedded authentication messages

image $I_{recovery}$

$$\begin{cases} I_{recovery}(x, y) = I'(x, y), & \text{if } TLM(k) == 1 \\ I_{recovery}(x, y + 1) = I'(x, y + 1), & \text{if } TLM(k) == 1 \\ I_{recovery}(x, y) = RM(k), & \text{if } TLM(k) == 0 \\ I_{recovery}(x, y + 1) = RM(k), & \text{if } TLM(k) == 0 \end{cases} \quad (7)$$

Where $x = \frac{k-1}{N/2} + 1$, $1 \leq x \leq M$, $y = 2 \times ((k - 1) \bmod (N/2) + 1) - 1$, $1 \leq y \leq N$ and $k = 1, 2, \dots, M \times \frac{N}{2}$

For a tampered RGB image I' to recover the RGB image, before executing Algorithm 4, first, the proposed scheme converts the image I' from RGB color space to YUV color space and gains I'_Y , I'_U and I'_V . Using Algorithm 4 input I'_Y , I'_U and I'_V separately, then obtain $I_{Yrecovery}$, $I_{Urecovery}$ and $I_{Vrecovery}$, with which we can convert the channels $I_{Yrecovery}$, $I_{Urecovery}$ and $I_{Vrecovery}$ from YUV color space to RGB color space to get color image $I_{recovery}$.

Table 2 shows the tampered images and the authentication results.

TABLE 1. PSNR of the original images compared with corresponding authentication messages embedded images

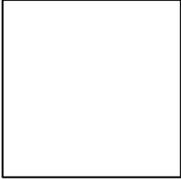
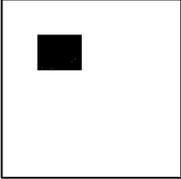
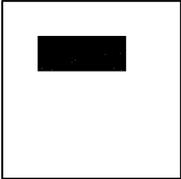
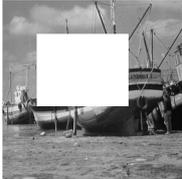
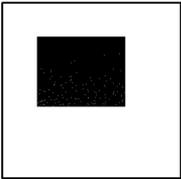
Grey Image	PSNR	Color Image	PSNR
(a) Lena	38.5970	(e) Lena	38.3129
(b) Ships	38.5328	(f) Girl	38.1553
(c) Baboon	38.9158	(g) Jet	37.7868
(d) Barbara	38.4804	(h) Boat	38.1075

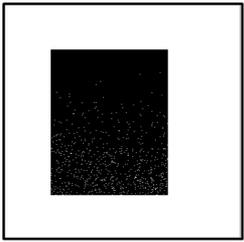
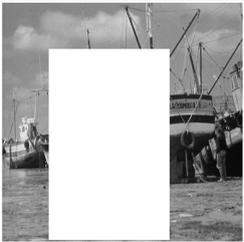
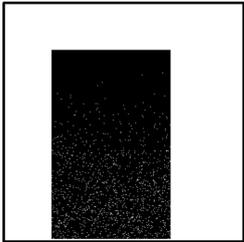
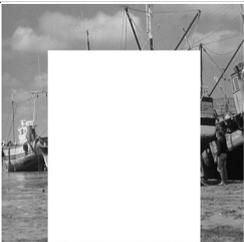
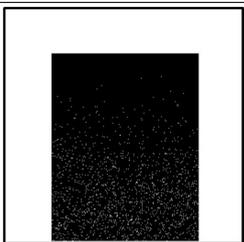
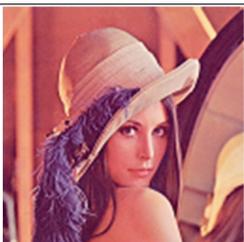
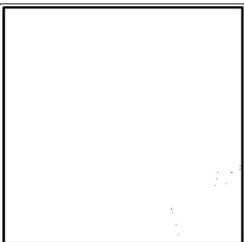
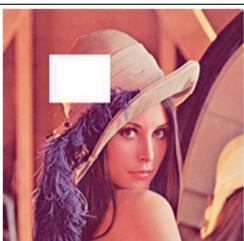
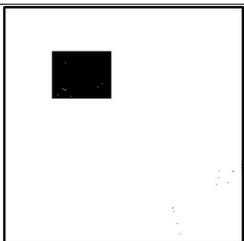
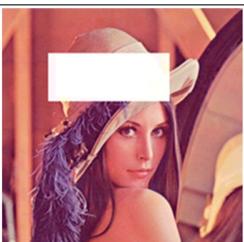
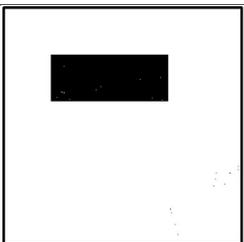
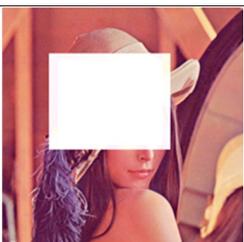
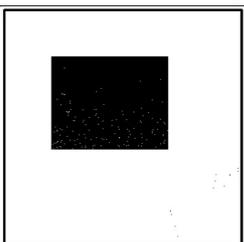
4. **Experimental Results.** The purpose of our scheme is to locate tamper regions more accurately. To accomplish this purpose, the proposed scheme uses the rehashing method to authenticate the integrity of an image's content.

To prove the performance of the proposed scheme, four 512×512 gray images and four 512×512 color images as shown in Fig. 5, were used as original images for comparisons of tamper detection and image recovery. Corresponding images which embedded authentication messages are shown in Fig. 6.

Table 1 shows the PSNR of the original images compared with corresponding authentication messages embedded images.

TABLE 2. Tampered images and their corresponding tamper detection locating image and recovery image

Attack	Tamper Image	Tamper Detection	Recovery Image
(a) no tamper			
(b) 5% modification			
(c) 10% modification			
(d) 20% modification			

(e) 30% modification			
(f) 40% modification			
(g) 50% modification			
(h) no tamper			
(i) 5% modification			
(j) 10% modification			
(k) 20% modification			

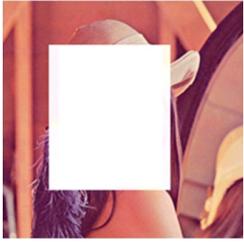
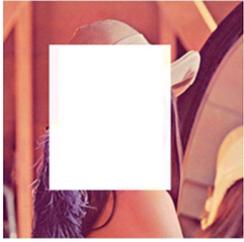
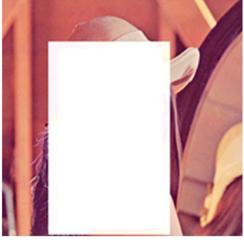
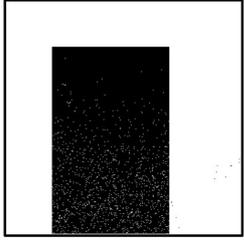
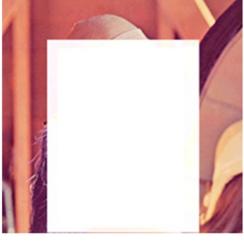
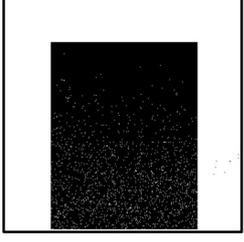
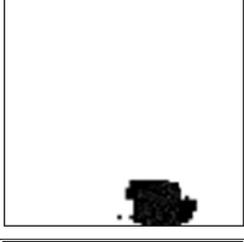
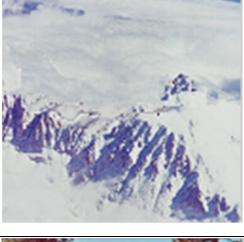
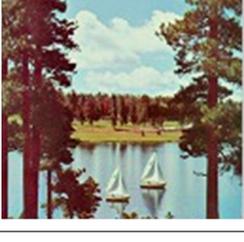
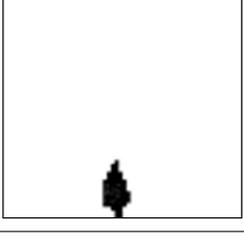
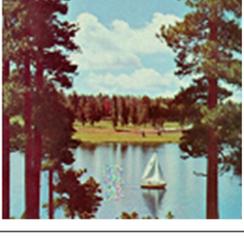
(l) 30% modification			
(m) 40% modification			
(g) 50% modification			
(p) special-shaped modification			
(q) special-shaped modification			
(r) special-shaped modification			

Table 3 uses statistical results to show tamper detection accuracy. The parameter N_t denotes the number of actually tampered pixels, N_{fd} denotes the number of tampered pixels detected as valid, and N_{fa} denotes the number of intact pixels detected as invalid. The total number of tampered pixels that are falsely detected is $N_{fd} + N_{fa}$. Then, the tamper detection probability (TDP), the false detection probability (FDP), false alarm probability (FAP) and average detection probability (ADP) are defined as Equations (8),

(9), (10) and (11), respectively

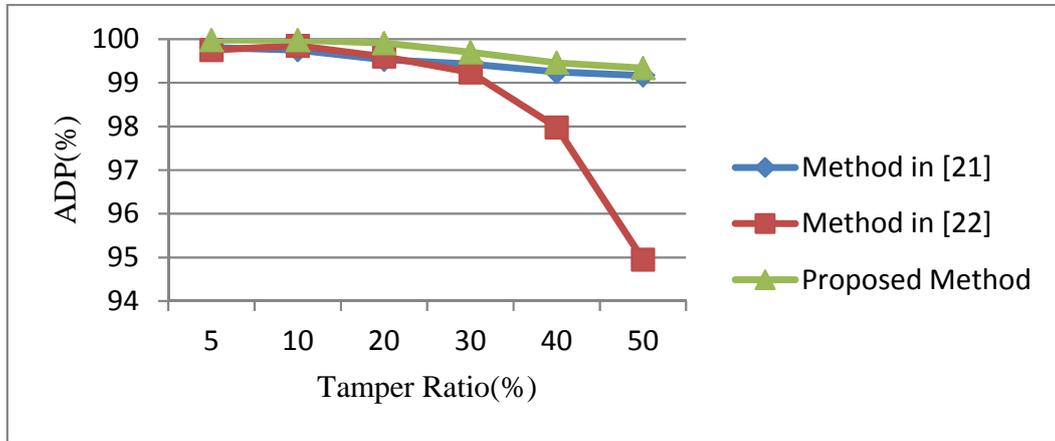
$$\text{TDP: } R_{td} = \left(\frac{N_t - N_{fd}}{N_t} \right) \times 100\%. \quad (8)$$

$$\text{FDP: } R_{fd} = \left(\frac{N_{fd}}{M \times N} \right) \times 100\%. \quad (9)$$

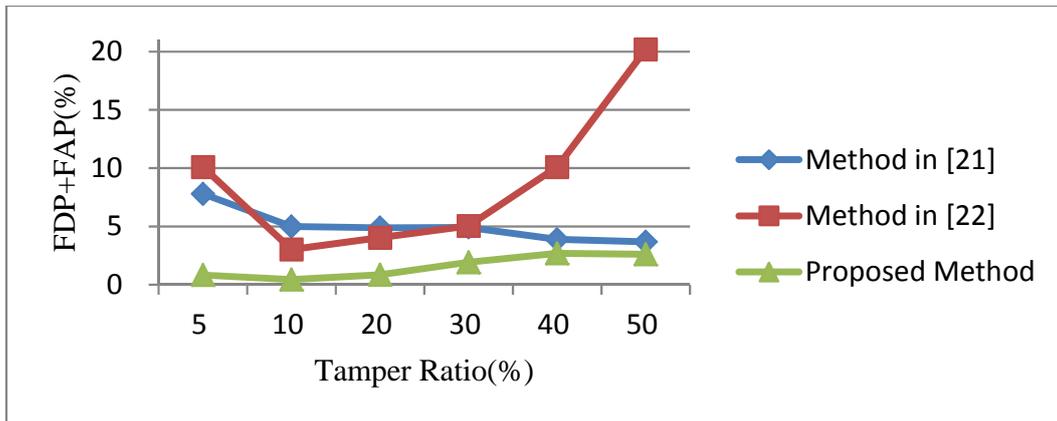
$$\text{FAP: } R_{fa} = \left(\frac{N_{fa}}{M \times N} \right) \times 100\%. \quad (10)$$

$$\text{ADP: } R_{ad} = \left(1 - \frac{R_{fd} + R_{fa}}{2} \right) \times 100\%. \quad (11)$$

The FDP and FAP are detection failing rates, and the ideal cases would be nearly 0%. The TDR and ADP are detection success rates and the ideal cases would be nearly 100%.



(a)



(b)

FIGURE 7. ADP and total false detection ratio comparison under the content tampering attack: (a) ADP; (b) FDP+FAP

Fig. 7 shows the results of comparing our proposed scheme with those of Kim et al's scheme [21] and He et al's scheme [22]. Using the rehashing method, the proposed scheme decreased collisions, produced low detection failing rates and provided high detection success rates, all of which are significant advantages.

TABLE 3. Statistics of experimental results

Tamper rate (Image size 512×512)	No. of tampered pixels	No. of correctly detected pixels (R_{td})	N_{fd}	N_{fa}	FDP (%)	FAP (%)	ADP (%)
Table 1(b) 5% modification	13287	13176 (99.16%)	111	0	0.0423	0	99.98
Table 1.(c) 10% modification	26471	26352 (99.55%)	119	0	0.0454	0	99.98
Table 1.(c) 10% modification	26471	26352(99.55%)	119	0	0.0454	0	99.98
Table 1.(d) 20% modification	52685	52232(99.14%)	452	0	0.1724	0	99.91
Table 1.(e) 30% modification	82497	80906(98.07%)	1591	0	0.6069	0	99.70
Table 1.(f) 40% modification	105370	102524(97.30%)	2846	0	1.0857	0	99.46
Table 1.(g) 50% modification	131610	128156(97.38%)	3454	0	1.3176	0	99.34
Table 1.(h) no tamper	0	0	0	26	0	0.0099	99.995
Table 1.(i) 5% modification	13287	13166(99.09%)	121	26	0.0462	0.0099	99.97
Table 1.(j) 10% modification	26471	26342(99.51%)	129	26	0.0492	0.0099	99.97
Table 1.(k) 20% modification	52685	52206(99.09%)	479	26	0.1827	0.0099	99.95
Table 1.(l) 30% modification	82497	80880(98.04%)	1617	26	0.6168	0.0099	99.69
Table 1.(m) 40% modification	105370	102524(97.30%)	2872	26	1.0956	0.0099	99.45
Table 1.(n) 50% modification	131610	128130(97.36%)	3480	16	1.3275	0.0061	99.33
Table 1(o) special-shaped modification	2752	2727(99.09%)	25	0	0.0001	0	99.995
Table 1(p) special-shaped modification	9488	9344(98.48%)	144	0	0.0005	0	99.975
Table 1(q) special-shaped modification	1072	1040	32	0	0.0001	0	99.995

5. **Conclusions.** An effective image tamper localization and self-recovery scheme that uses rehashing technology is proposed in this paper. The proposed scheme utilizes a rehashing model to overcome easy collisions of the numerous metadata in key space, thereby innovatively enhancing its effectiveness against attacks. This scheme embeds image feature values in the LSBs of the original image to protect the content of the image. The

proposed scheme was tested with grayscale and color images to test its performance. The experimental results demonstrated that the proposed scheme has good performance in tamper location and allows image recovery with an acceptable visual quality when there was up to 50% content tampering. Also, the experimental results verified that the proposed method can be used for image authentication applications

REFERENCES

- [1] C. Y. Lin, S. F. Chang, SARI: Self-authentication-and-recovery Image Watermarking System *Proceedings of the ninth ACM international conference on Multimedia ACM*, pp. 628-629, 2001.
- [2] R. Chamlawi, A. Khan, I. Usman, Authentication and Recovery of Images Using Multiple Watermarks , *Computers & Electrical Engineering* vol. 36, no.3 pp. 578-584, 2010.
- [3] R. Chamlawi, A. Khan, Digital Image Authentication And Recovery: Employing Integer Transform Based Information Embedding And Extraction *Information Sciences* vol.180, no.24, pp. 4909-4928, 2010.
- [4] S. Bravo-Solorio, A. K. Nandi, Secure Fragile Watermarking Method for Image Authentication with Improved Tampering Localisation and Self-Recovery Capabilities *Signal Processing* vol.91, no.4, pp.728-739, 2011.
- [5] L. Yang, R. Ni, Y. Zhao, Segmentation-based Image Authentication and Recovery Scheme Using Reference Sharing Mechanism *2012 International Conference on Industrial Control and Electronics Engineering ICICEE 2012* pp. 863-866, 2012.
- [6] R. Ullah, A. Khan, A. S. Malik, Dual-purpose Semi-fragile Watermark: Authentication and Recovery of Digital Images , *Computers & Electrical Engineering* vol. 39, No 7 pp. 2019-2030, 2013.
- [7] L. Rosales-Roldan, M. Cedillo-Hernandez, M. Nakano-Miyatake, H. Perez-Meana, B. Kurkoski, Watermarking-based Image Authentication with Recovery Capability Using Halftoning Technique *Signal Processing: Image Communication* vol.28, no.1 pp. 69-83, 2013.
- [8] P. L. Lin, C. K. Hsieh, P. W. Huang, A Hierarchical Digital Watermarking Method for Image Tamper Detection and Recovery *Pattern Recognition* vol.38, no.12, pp. 2519-2529 2005.
- [9] C. C. Chang, Y. H. Fan, W. L. Tai, Four-scanning Attack on Hierarchical Digital Watermarking Method for Image Tamper Detection and Recovery , *Pattern Recognition* vol.41, no.2 pp.654-661, 2008.
- [10] S. Rawat, B. Raman, A Chaotic System Based Fragile Watermarking Scheme for Image Tamper Detection *AEU-International Journal of Electronics and Communications* vol.65, no.10 pp. 840-847, 2011.
- [11] X. Tong, Y. Liu, M. Zhang, Y. Chen, A Novel Chaos-Based Fragile Watermarking for Image Tampering Detection and Self-Recovery *Signal Processing: Image Communication* vol. 28, no. 3, pp. 301-308, 2012.
- [12] C. W. Lee, W. H. Tsai, A Data Hiding Method Based on Information Sharing via PNG Images for Applications of Color Image Authentication and Metadata Embedding , *Signal Processing*, vol. 93., No 7, pp.2010-2025, 2013
- [13] R. Venkatesan, S. M. Koon, M. H. Jakubowski, P. Moulin, Robust Image Hashing *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 664-666, 2000
- [14] J. Fridrich, M. Goljan, Robust Hash Functions for Digital Watermarking *Proceedings of IEEE International Conference on Information Technology: Coding and Computing*, pp. 178-183, 2000
- [15] M. Tagliasacchi, G. Valenzise, S. Tubaro, Hash-Based Identification of Sparse Image Tampering *IEEE Transactions on Image Processing*, vol.18, no.11, pp. 2491-2504, 2009.
- [16] F. Ahmed, M. Y. Siyal, V. Uddin Abbas, A Secure and Robust Hash-Based Scheme for Image Authentication *Signal Processing*, vol. 90, no. 5, pp. 1456-1470, 2010.
- [17] Y. Lei, Y. Wang, J. Huang, Robust Image Hash in Radon Transform Domain for Authentication *Signal Processing: Image Communication*, vol. 26, no. 6, pp. 280-288, 2011
- [18] Y. Zhao, S. Wang, X. Zhang, H. Yao, Robust Hashing for Image Authentication Using Zernike Moments and Local Features *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 55-63, 2013.
- [19] R. Sprugnoli, Perfect Hashing Functions: a Single Probe Retrieving Method for Static Sets *Communications of the ACM*, vol. 20, no. 11, pp. 841-850, 1977.
- [20] M. W. Du, T. M. Hsieh, K. F. Jea, D. W. Shieh, The Study of a New Perfect Hash Scheme *IEEE Transactions on Software Engineering* , vol. SE-9, no. 3, pp. 305-313, 1983

- [21] K. S. Kim, M. J. Lee, J. M. Lee, T. W. Oh, H. Y. Lee, Region-based Tampering Detection and Recovery Using Homogeneity Analysis in Quality-sensitive Imaging *Computer Vision and Image Understanding* vol.115, no.9 pp.1308-1323, 2011.
- [22] H. He, F. Chen, H. M. Tai, T. Kalker, J. Zhang, Performance Analysis of a Block-neighborhood-based Self-recovery Fragile Watermarking Scheme *IEEE Transactions on Information Forensics and Security*, vol.7, no.1 pp.185-196, 2012.