A REVERSIBLE IMAGE AUTHENTICATION METHOD FREE FROM LOCATION MAP AND PARAMETER MEMORIZATION

Seungwu HAN, Masaaki FUJIYOSHI, and Hitoshi KIYA

Department of Information and Communication Systems Engineering
Tokyo Metropolitan University
Tokyo, Japan
E-mail: han-seungwu@sd.tmu.ac.jp, mfujiyoshi@m.ieice.org, kiya@eei.metro-u.ac.jp

ABSTRACT

This paper proposes a novel reversible image authentication method that requires neither location map nor memorization of parameters. The proposed method detects image tampering and further localizes tampered regions. Though this method once distorts an image to hide data for tamper detection, it recovers the original image from the distorted image unless no tamper is applied to the image. The method extracts hidden data and recovers the original image without memorization of any location map that indicates hiding places and of any parameter used in the algorithm. This feature makes the proposed method practical. Simulation results show the effectiveness of the proposed method.

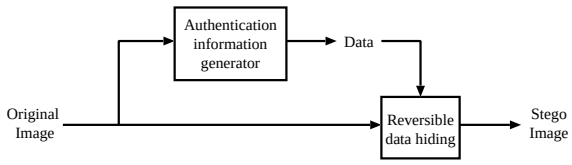
Keywords: tamper detection, reversible data hiding, lossless watermarking, parameter memorization-free, location map-free

1. INTRODUCTION

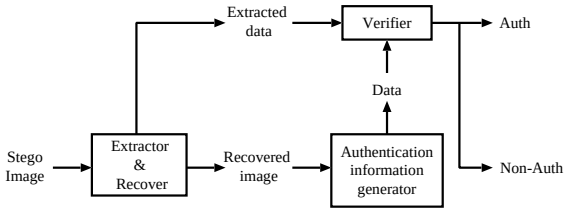
Currently, powerful computers and smart software make modification of digital images very easy. Moreover, digital images are often widely distributed through the Internet. It is, therefore, very simple for malicious users to make any tampered image available to others. Image authentication that insures digital image integrity has therefore became a major issue. There are two approach for image authentication; the classical that stores and transmits the signature of an image separately from the image [1–4] and the data hiding based [5–12]. This paper focuses the latter.

In an image authentication method based on data hiding techniques, the signature or the feature of the image is embedded into and extracted from the image, where an image conveying data is referred to as a *stego* image. Though the original image is distorted to convey hidden data, several methods [9–12] can recover the original image after authentication, i.e., *reversible*. A reversible authentication method that only memorizes one parameter and does not has to memorize hiding positions [12] is practical. However, the only one parameter is image dependent.

This paper proposes a novel practical reversible authentication method. The proposed method does not memorize any parameter to extract hidden data as well as it does not require any location map indicating hiding places. That is, the proposed method extracts hidden data from a stego image without any image dependent information. This feature makes the proposed method practical.



(a) Data generation and data hiding.



(b) Data extraction, image recovery, and verification

Fig. 1. Block diagram of typical reversible image authentication.

This paper consists of five sections. In Sect. 2, a brief overview of the conventional methods [9–12] are given. The novel method is proposed in Sect. 3. Experimental results is given in Sect. 4. Conclusions are drawn in Sect. 5.

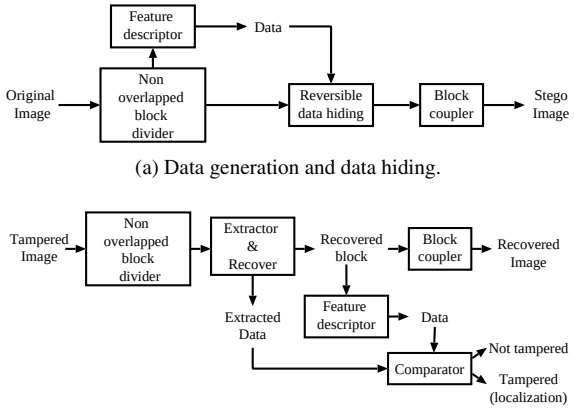
2. BACKGROUND

This section gives the framework of reversible image authentication and particular conventional method.

2.1. Reversible Image Authentication

A typical reversible image authentication method is shown in Fig. 1. A reversible image authentication method has two steps in the process to generate stego images as shown in Fig. 1 (a): an authentication data generation and reversible data hiding. That is, in part (a), the generated data are hidden to the image in a reversible mannar.

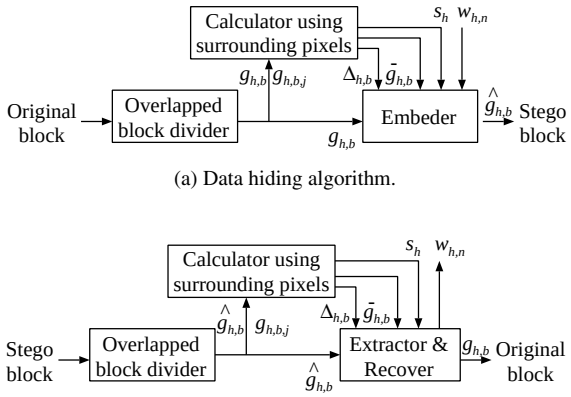
The counterpart process is shown in Fig. 1 (b): an extraction of hidden data, recovery of the original image, and integrity verification. In part (b), authentication data are generated from the recovered image, and are compared with



(a) Data generation and data hiding.

(b) Data extraction, image recovery, and verification.

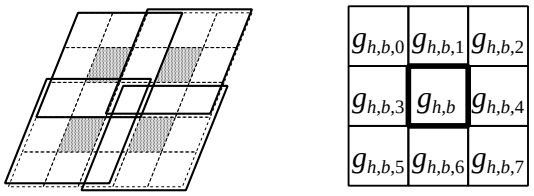
Fig. 2. The proposed method.



(a) Data hiding algorithm.

(b) Hidden data extraction and image recovery algorithm.

Fig. 3. Reversible data hiding in the proposed method.



(a) Placement of subblocks.

(b) A subblock.

Fig. 4. Subblock for data hiding.

the extracted data. If two data are identical, it determines that the image has not been modified, and the recovered image is labeled as genuine.

2.2. Conventional Methods and Those Problems

This section briefly mentions the typical conventional methods; they are compression-based methods [9, 10] and non-compression-based methods [11, 12].

Among compression-based methods, one method [9] reversibly compresses the least significant bitplane among bitplanes meeting the criteria that the rooms generated by compression is larger than the length of the hash of the whole image. The other method [10] uses arithmetic encoder to reversibly compresses less significant bitplanes of the image by utilizing more significant bitplanes to create the rooms to hide data block-by-block. This method repeats such embedding process multiple times. They use compression techniques that usually cost highly.

On the other hand, the non-compression-based methods [11, 12] are based on the block-based reversible data hiding scheme [13, 14] that uses statistics of pixels. They do not require any location map for hidden data extraction and image recovery and they only memorize one parameter. The parameter, however, is image dependent, so these methods require different parameters for different images.

In the next section, a practical reversible authentication method is proposed.

3. PROPOSED METHOD

This section proposes an image authentication method that improves the practicality.

As in Fig. 2, the proposed method consists of two steps: Feature generation and reversible data hiding in Step (a), and hidden data extraction, image recovery, and integrity verification in Step (b). Each of them is described in the following subsections, where an original image is assumed to be an $X \times Y$-sized grayscale image in which each pixel has 2^K levels from zero to $2^K - 1$. An original image is divided into H non-overlapped blocks which each block consists of $H_X \times H_Y$ pixels, where $H = XY/H_X H_Y$. An image feature descriptor that generates an N-bits length data strings is applied to each block, and the generated data string is hidden to the block from that data string is generated. That is, integrity is verified in terms of blocks.

3.1. Image Feature Description

An image feature descriptor generates a data string that is to be reversibly hidden to the input image. Though any arbitrary descriptor is able to be used, an one-way hash function is employed in this paper to improve the reliability of tamper detection and to guarantee the *payload* (length) of the description. An one-way hash function that generates N-bits length hash is applied to each block. The hash for the h-th block is represented by $\mathbf{w}_h = \left\{ w_{h,n} \middle| w_{h,n} \in \{0,1\}, n = 0, 1, \ldots, N-1 \right\}$, where $h = 0, 1, \ldots, H-1$.

3.2. Reversible Data Hiding

This section describes the non-compression-based data hiding algorithm (Fig. 3 (a)) and the corresponding hidden data extraction and image recovery algorithm (Fig. 3 (b)) are described. It also describes parameters that are used in both algorithms. It is noteworthy that the proposed method does not have to memorize any parameter.

The proposed method reversibly hides N-bits hash $\mathbf{w}_h$ to the corresponding h-th block based on the reversible data hiding scheme [13, 14]. In a data hiding step, a bit of a hash is hidden in terms of overlapped subblocks as shown in Fig. 4 (a). Pixel $g_{h,b}$ is the center pixel of the b-th overlapped subblock in the h-th block as shown in Fig. 4 (b), where

$b = 0, 1, \ldots, B_h - 1$. Since the proposed method hides $w_{h,n}$ to $g_{h,b}$, B represents the ideal *capacity* of the h-th block. Ideal capacity B is given by

$$B = \left\lfloor \frac{H_X - 1}{2} \right\rfloor \left\lfloor \frac{H_Y - 1}{2} \right\rfloor \quad \text{[bits]}, \tag{1}$$

where $\lfloor p \rfloor$ rounds real-value p to the nearest integer towards minus infinity.

This data hiding uses one parameter per non-overlapped block that is commonly used in embedding and extraction processes. Therefore, derivation of the parameter is firstly described in the next section, Then, embedding and extraction algorithms that both of them use the derived parameter are described.

3.2.1. Parameter Derivation

The algorithm to derive parameter s_h for the h-th block from an original image consists of the coarse part and the refine part. The coarse part is following.

1. $b := 0$ and $\beta_h := 0$.
2. In the b-th subblock of the h-th block, the average value is obtained from the surrounding pixels, $g_{h,b,j}$ ($j = 0, 1, \ldots, 7$), by Eq. (2). Difference between center pixel $g_{h,b}$ and average $\bar{g}_{h,b}$ is also derived by Eq. (3).

$$\bar{g}_{h,b} = \left\lfloor \frac{1}{8} \sum_{j=0}^{7} g_{h,b,j} \right\rfloor, \tag{2}$$

$$d_{h,b} = g_{h,b} - \bar{g}_{h,b}. \tag{3}$$

3. $\Delta_{h,b}$ is obtained by Eq. (4).

$$\Delta_{h,b} = \begin{cases} g_{\max,h,b} - \bar{g}_{h,b}, & d_{h,b} \geq 0 \\ g_{\min,h,b} - \bar{g}_{h,b}, & d_{h,b} < 0 \end{cases}, \tag{4}$$

$$g_{\max,h,b} = \max_j g_{h,b,j}, \tag{5}$$

$$g_{\min,h,b} = \min_j g_{h,b,j}. \tag{6}$$

4. Parameter $s_{h,b}$ that is a candidate of s_h is derived by Eq. (7).

$$s_{h,b} = \begin{cases} \left|\Delta_{h,b}\right|, & \bar{g}_{h,b} + 2d_{h,b} < 0 \\ & \text{or} \quad 2^K - 2 < \bar{g}_{h,b} + 2d_{h,b} \\ \infty, & \text{others} \end{cases}, \tag{7}$$

   where K represents the quantization bits for pixel values, i.e., $K = 8$ for eight-bits quantized grayscale images.
5. If $s_{h,b} = \left|\Delta_{h,b}\right|$, $\beta_h := \beta_h + 1$.
6. $b := b + 1$. Continue to Step 2 unless $b = B$.
7. The smallest of $s_{h,b}$ becomes s_h. That is,

$$s_h = \min_b s_{h,b}. \tag{8}$$

The refine part that adequately adjusts s_h derived by the coarse part to the h-th block in terms of capacity β_h is following.

1. $t := s_h$.
2. If $\beta_h = N$, proceed to Step 9.
3. If $\beta_h > N$, $s_h := s_h - 1$ and proceed to the next step. If $\beta_h < N$, $s_h := s_h + 1$ and proceed to the next step.
4. If $s_h = t$, proceed to Step 9. If $s_h = 0$, $s_h := 1$ and proceed to Step 9. If $s_h = 2^K$, $s_h := 2^K - 1$ and proceed to Step 9.
5. $b := 0$ and $\beta_h := 0$.
6. If $0 \leq \bar{g}_{h,b} + 2d_{h,b} \leq 2^K - 2$, $\beta_h := \beta_h + 1$.
7. $b := b + 1$. Continue to Step 6 unless $b = B$.
8. Continue to Step 2.
9. Smallest s_h satisfying $\beta_h \geq N$ is obtained.

3.2.2. Hiding Algorithm

As in Fig. 3 (a), this algorithm decides whether $g_{h,b}$ is embeddable or not based on parameter s_h. The following algorithm is applied to the h-th block to hide N-bits data $\mathbf{w}_h$.

1. $b := 0$ and $n := 0$.
2. By Eq. (9), $\hat{g}_{h,b}$, the pixel with hidden data, is derived from embeddable $g_{h,b}$.

$$\hat{g}_{h,b} = \begin{cases} \bar{g}_{h,b} + 2d_{h,b} + w_{h,n}, & \left|\Delta_{h,b}\right| < s_h \\ g_{h,b}, & \text{others} \end{cases}. \tag{9}$$

3. If $\left|\Delta_{h,b}\right| < s_h$, $n := n + 1$.
4. $b := b + 1$. Continue Step 2 until $b = B$.
5. The h-th stego block conveying N-bits data $\mathbf{w}_h$ is obtained.

All H of stego blocks are combined to form a stego image. It is noteworthy that parameter s_h used in this algorithm is oblivious. That is, s_h is not memorized in the proposed method.

3.2.3. Hidden Data Extraction and Image Recovery Algorithm

The algorithm to extract hidden data $\mathbf{w}_h$ and restore the original image block consists of two parts; one is parameter estimation and the other is hidden data extraction and image recovery. The parameter estimation part is following.

1. $s_h := 1$.
2. $b := 0$ and $\beta_h := 0$.

3. $\Delta_{h,b}$ is obtained by Eq. (10).

$$\Delta_{h,b} = \begin{cases} g_{\max,h,b} - \bar{g}_{h,b}, & \hat{g}_{h,b} - \bar{g}_{h,b} \geq 0 \\ g_{\min,h,b} - \bar{g}_{h,b}, & \hat{g}_{h,b} - \bar{g}_{h,b} < 0 \end{cases}. \quad (10)$$

4. If $\left|\Delta_{h,b}\right| < s_h$, $\beta_h := \beta_h + 1$.

5. $b := b + 1$. Continue to Step 3 unless $b = B$.

6. If $\beta_h < N$, $s_h := s_h + 1$ and continue to Step 2.

7. Smallest s_h satisfying $\beta_h \geq N$ is estimated.

The hidden data extraction and image recovery part using estimated s_h is following.

1. $b := 0$ and $n := 0$.

2. $\Delta_{h,b}$ is obtained by Eq. (10).

3. Data bit $w_{h,n}$ is extracted by the following equation, if $\left|\Delta_{h,b}\right| < s_h$.

$$w_{h,n} = (\hat{g}_{h,b} - \bar{g}_{h,b}) \bmod 2. \quad (11)$$

4. Pixel $g_{h,b}$ of the image is restored by Eq. (12).

$$g_{h,b} = \begin{cases} \frac{\hat{g}_{h,b} + \bar{g}_{h,b} - w_{h,b}}{2}, & \left|\Delta_{h,b}\right| < s_h \\ \hat{g}_{h,b}, & \text{others} \end{cases}. \quad (12)$$

5. If $\left|\Delta_{h,b}\right| < s_h$, $n := n + 1$.

6. $b := b + 1$. Continue to Step 2 unless $b = B$.

7. N-bits data sequence $\mathbf{w}_h$ and the h-th recovered image block are obtained.

All H of recovered blocks are combined to form the recovered image.

As described above, this algorithm does not memorize any parameter nor location map.

3.2.4. Integrity Verification

Integrity verification process in the proposed method is comparing two image features as shown in Fig. 2 (b). One is the extracted feature description and the other is the feature description generated from the recovered image. If both descriptions are the same among all H blocks, the image is not modified. On the other hand, one is different from the other in the h-th block, the h-th block of the image is altered. That is, the proposed method detects tampering and also localizes the altered region by the unit of block.

3.3. Features

This section describes two major features of the proposed method that contributes practical image authentication.

3.3.1. Location Map Free

This feature is mainly from the used reversible data hiding scheme [13, 14]. The proposed method leaves $g_{h,b}$ as is, unless $g_{h,b}$ is embeddable, as shown in Eq. (9). Thus, a location map indicating hiding positions is generally required to extract hidden data, but the proposed method does not use any location map. From the embedding algorithm, if $\left|\Delta_{h,b}\right| < s$, original pixel $g_{h,b}$ is determined to be embeddable by Eq. (9). This $\Delta_{h,b}$ is computable from a stego image, but it varies according to $d_{h,b}$, as shown in Eq. (4). This $d_{h,b}$ depends on original pixel $g_{h,b}$, as shown in Eq. (3). However, the proposed method solves this problem as follows. From Eq. (9),

$$\hat{g}_{h,b} - \bar{g}_{h,b} = \begin{cases} \bar{g}_{h,b} + 2d_{h,b} + w_{h,n} - \bar{g}_{h,b} \\ \qquad = 2d_{h,b} + w_{h,n} & \left|\Delta_{h,b}\right| < s_h \\ g_{h,b} - \bar{g}_{h,b} = d_{h,b}, & \text{others} \end{cases}. \quad (13)$$

Under the conditions that $w_{h,n} \in \{0, 1\}$ and $d_{h,b}$ is an integer,

$$\begin{cases} 2d_{h,b} + w_{h,n} \geq 0, & d_{h,b} \geq 0 \\ 2d_{h,b} + w_{h,n} < 0, & d_{h,b} < 0 \end{cases}. \quad (14)$$

From Eqs. (13) and (14),

$$\begin{cases} \hat{g}_{h,b} - \bar{g}_{h,b} \geq 0, & d_{h,b} \geq 0 \\ \hat{g}_{h,b} - \bar{g}_{h,b} < 0, & d_{h,b} < 0 \end{cases}, \quad (15)$$

and this property of the proposed method is used in Eq. (11) to determine whether $\hat{g}_{h,b}$ conveys $w_{h,n}$.

3.3.2. Parameter Oblivion

In the proposed method, both the data hiding algorithm and the hidden data extraction and image recovery algorithm use parameter s_h. Though s_h varies dependently to the image and the block, s_h is not transmitted from the data hiding algorithm to the hidden data extraction and image recovery algorithm in the proposed method. This feature is realized by the follows. The proposed method set s_h as the smallest value that satisfies $\beta_h \geq N$, i.e., the capacity of the h-th block is equal or larger than the payload of hidden data. From the parameter derivation algorithm, it is clear that s_h in an integer satisfying $1 \leq s_h \leq 2^K - 1$. These properties make the estimation of s_h feasible. As described in the parameter estimation algorithm in Sect. 3.2.3, it simply estimates s_h by comparing β_h and N. As mentioned above, the smallest s_h satisfying $\beta_h geN$ is very s_h used in the data hiding algorithm, so the estimation of s_h is simple and feasible.

4. EXPERIMENTAL RESULTS

The proposed method was evaluated with 512×512-sized, i.e., $X = Y = 512$, grayscale images from CIPR-RPI [15] that are shown in Fig. 5. In this evaluation, SHA-256 [16] was used as the one-way hash function for describing image feature and SHA-256 gives 256-bits has, i.e., $N = 256$. Conditions are summarized in Table 1.

(a) Airplane.

(b) Lena.

(c) Peppers.

(d) Sailboat.

(e) Tiffany.

Fig. 5. 512×512-sized grayscale images for evaluation from CIPR-RPI [15].

Table 1. Conditions.

(a) Fundamenlta conditions.

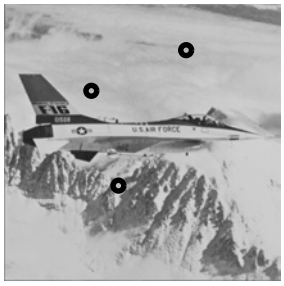
Image size $X \times Y$	512×512 [pixels]
Image feature descriptor	SHA-256 [16]
Description length N	256 [bits]

(b) Block configurations.

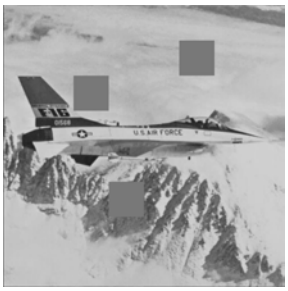
Block size $H_X \times H_Y$ [pixels]	Ideal capacity per block B [bits]	The number of blocks H	Payload per image NH [bits]
512×512	65025	1	256
256×256	16129	4	1024
128×128	3969	16	4096
64×64	961	64	16384

Figure 6 (a) is the stego image generated by the proposed method from the image shown in Fig. 5 (a) with 64×64-sized blocks. Figure 6 (b) shows a tampered image in which three circles indicate points different from the stego image. The proposed method correctly detects image tampering and localizes all tampered regions as shown in Fig. 6 (c). Figure 7 shows the tamper detection and localization results of the proposed method using various size blocks in which the tampered regions are as the same as that in Fig. 6 (b).

(a) Stego image (PSNR: 52.2 dB).

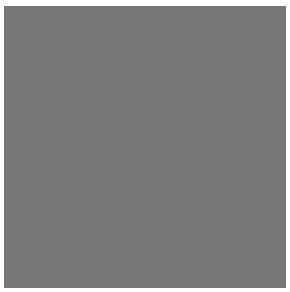


(b) Tampered image.

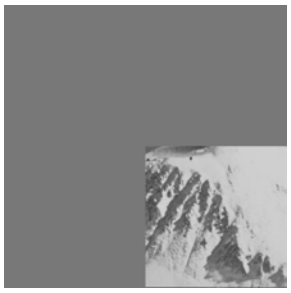


(c) Localized tampered regions.

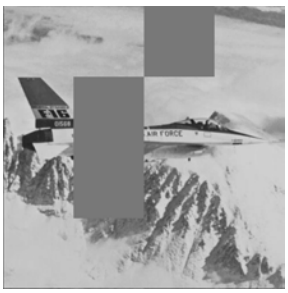
Fig. 6. Results of tamper detection and localization ($H = 64$).



(a) $H = 1$.



(b) $H = 4$.



(c) $H = 16$.

Fig. 7. Tamper detection and localization for various H.

Figure 8 shows the PSNR's of stego images generated by the proposed method. All stego images have the quality superior than 45 dB in terms of the PSNR, though it conveys

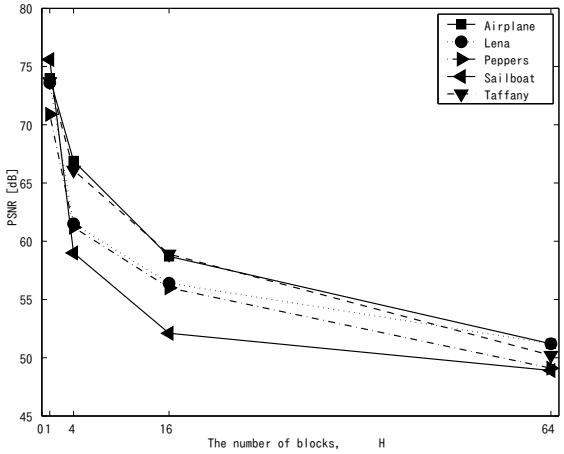


Fig. 8. The PSNR's of stego images versus the number of blocks, H.

16384 bits hidden data to localize the tampered regions by the unit of 64×64-sized block. If the application only requires the method to detect tampering and does not require localization of altered regions, i.e., $H = 1$ in the proposed method, the PSNR's of stego images are over 70 dB.

5. CONCLUSIONS

This paper has proposed a novel reversible image authentication method. The proposed method extracts hidden data and recovers the original image without memorizing any image dependent information; a location map and parameter. The method, thus, is practical. This paper only focuses the fundamental of the proposed method, so using different size and shape of overlapped subblocks [14], encipherment of feature description [12], and hierarchical data hiding [12] are applicable to the proposed method to improve the efficiency and security.

REFERENCES

[1] M. Schneider and S.F. Chang, "A robust content based digital signature for image authentication," Proc. IEEE Int. Conf. Image Process., vol.3, pp.227–230, Switzerland, Sept. 1996.

[2] R. Gennaro and P. Rohatgi, "How to sign digital streams," Proc. IACR Annual Int. Cryptology Conf., pp.180–197, Santa Barbara, CA, the U.S., Aug. 1997.

[3] S. Bhattacharjee and M. Kutter, "Compression tolerant image authentication," Proc. IEEE Int. Conf. Image Process., pp.435–439, Chicago, IL, the U.S., Oct. 1998.

[4] H. Kobayashi and H. Kiya, "Robust image authentication using hash function," Proc. IEEE Region 10 Conf., Chiang Mai, Thailand, Nov. 2004.

[5] P.W. Wong, "A public key watermark for image verification and authentication," Proc. IEEE Int. Conf. Image Process., vol.1, pp.425–429, Chicago, IL, the U.S., Oct. 1998.

[6] C.Y. Lin and S.F. Chang, "Semi-fragile watermarking for authenticating JPEG visual content," Proc. SPIE, vol.3971, pp.140–151, San Jose, CA, the U.S., Jan. 2001.

[7] J. Fridrich, "Security of fragile authentication watermarks with localization," Proc. SPIE, vol.4675, pp.691–700, San Jose, CA, the U.S., Jan. 2002.

[8] J. Wu, B.B. Zhu, S. Li, and F. Lin, "A secure image authentication algorithm with pixel-level tamper localization," Proc. IEEE Int. Conf. Image Process., vol.3, pp.1573–1576, Singapore, Oct. 2004.

[9] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," Proc. SPIE, vol.3971, pp.197–208, San Jose, CA, the U.S., Jan. 2001.

[10] M.U. Celik, G. Sharma, E. Saber, and A.M. Tekalp, "Lossless watermarking for image authentication: A new framework and an implementation," IEEE Trans. Image Process., vol.15, no.4, pp.1042–1049, Apr. 2006.

[11] S. Han, H.L. Jin, M. Fujiyoshi, and H. Kiya, "Lossless data hiding in the spatial domain for image tamper detection," Proc. IEEE Int. Sympo. Intelligent Signal Process. and Comm. Sys., pp.760–763, Yonago, Japan, Dec. 2006.

[12] S. Han, M. Fujiyoshi, and H. Kiya, "An efficient reversible image authentication method," IEICE Trans. Fundatmentals, vol.E91-A, no.8, pp.1907–1914, Aug. 2008.

[13] H.L. Jin, M. Fujiyoshi, and H. Kiya, "Lossless data hiding in the spatial domain for high quality images," IEICE Trans. Fundamentals, vol.E90-A, no.4, pp.771–777, Apr. 2007.

[14] M. Fujiyoshi, S. Sato, H.L. Jin, and H. Kiya, "A location-map free reversible data hiding method using block-based single parameter," Proc. IEEE Int. Conf. Image Process., vol.III, pp.257–260, San Antonio, TX, the U.S., Sept. 2007.

[15] Centre for Image Processing, Renesselaer Polytechnic Institute, "Still image and sequences," `http://www.cipr.rpi.edu`

[16] "Secure hash standard," NIST FIPS 180-2, Aug. 2002, `http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf`