Signal Processing 93 (2013) 2010–2025

Contents lists available at SciVerse ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A data hiding method based on information sharing via PNG images for applications of color image authentication and metadata embedding☆

Che-Wei Lee[a], Wen-Hsiang Tsai[a,b,*]

[a] Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan
[b] Department of Information Communication, Asia University, Taichung 41354, Taiwan

ARTICLE INFO

Article history:
Received 14 February 2012
Received in revised form
9 November 2012
Accepted 10 January 2013
Available online 4 February 2013

Keywords:
Data hiding
Secret sharing
Color image authentication
PNG image
Alpha channel

ABSTRACT

A new data hiding method via PNG images based on Shamir's (k, n)-threshold secret sharing scheme is proposed. The coefficients of the polynomial used by the Shamir's method are taken as carriers to carry given data. The data to be hidden then are transformed into partial shares and embedded into the alpha-channel plane of a cover PNG image. Undesired white noise created in the resulting stego-image is removed by skillfully mapping the computed share values into suitable ranges in the embedding process. An important application of the proposed method to color image authentication is also proposed, with detailed authentication algorithms for PNG images presented. Compared with existing methods, the proposed image authentication method possesses the merits of losslessness during image authentication, high sensitivity to image alterations, good tampering localization capability, and very low false acceptance and rejection ratios. Experimental results proving the effectiveness of the proposed methods are also included.

1. Introduction

Data hiding is a process of embedding information into a certain digital file that acts as a host. Through data hiding techniques, information such as authentication signals can be embedded into a digital file for the purpose of verifying the integrity or fidelity of the file [1–4]. In the application of copyright protection, an owner of a digital file can use data hiding techniques to embed a visible or invisible digital watermark into the file content to claim the ownership of the content [5–8]. Another application of data hiding is covert communication [9–10] in which people hide a secret message into a *cover file*, resulting in a *stego-file*; and a receiver of the latter can extract the hidden message from the stego-file to complete the communication.

In addition, *information sharing* was proposed to protect the security of concerned data by transforming a secret message into several *shares* which are then distributed to a number of participants to keep. Such a *secret sharing* scheme is useful for reducing the risk of incidental data loss and advantageous for keeping a balance among the participants: only when all the shares or a sufficient number of them are collected from the participants can the secret message be recovered correctly. This concept of secret sharing was proposed first by Shamir [11]. Conventionally, data hiding and information sharing are two irrelevant issues in the domain of information security.

☆ This work was supported partially by the NSC project No. 99-2631-H-009-001.

* Corresponding author at: National Chiao Tung University, Department of Computer Science and Information Engineering, Hsinchu 30010, Taiwan. Tel.: +886 3 5728368; fax: +886 3 5734935.
E-mail addresses: paradiserlee@gmail.com (C.-W. Lee), whtsai@cis.nctu.edu.tw (W.-H. Tsai).

0165-1684/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.sigpro.2013.01.009

In this study, a new data hiding method based on the technique of information sharing is proposed for hiding data into PNG (portable network graphics) images.

In the literature, many data hiding methods exploring the spatial domain and frequency domain of images [12–16] have been proposed. Bender et al. [12] proposed the technique of least-significant-bit (LSB) replacement, in which a secret message is embedded in the least significant bits of image pixel values. Mielikainen [13] proposed a modified LSB replacement method which embeds as many bits as the conventional method, but changes less pixel values. Yang et al. [14] proposed an adaptive *k*-LSB substitution method in which larger values of *k* are adopted in the edge areas of the cover image and smaller ones are used for the smooth areas. Wang et al. [15] transformed image block contents into coefficients in the frequency domain by the discrete cosine transform (DCT) and embedded secret bits by modifying the magnitude relations between the AC values of image blocks. Besides data embedding techniques using the DCT, the discrete wavelet transform (DWT) [17] and the discrete Fourier transform (DFT) [18,19] have also been used.

From another viewpoint, different *types* of images and files can be used as cover media for developing data hiding [20,21]. In [21], Lee and Wu proposed a lossless data hiding method for palette-based images, which adjusts palette colors and image data to embed secret data and side information for reconstruction of the original image content. Lee and Tsai [9] hid data into PDF files' characters by using special ASCII codes. Liu [10] made use of the change tracking function in Microsoft Word to hide data by a document degeneration technique.

In this paper, in addition to the aforementioned spatial domain, frequency domain and palettes of images for use in data hiding, we try to explore a new embeddable space for data hiding, aiming at providing more data hiding capacity, better quality of the resulting stego-image and stronger applicability. As a result, the PNG image with the alpha channel plane is found to be capable of meeting these requirements mentioned previously. Specifically, in the *information-sharing-based* data hiding method proposed in this study, a PNG image is used as the cover image in which the alpha-channel value of each pixel is set to be 255 initially. That is, the cover image is a totally transparent color one at the beginning of the proposed data hiding process. A data string to be hidden is transformed into shares by the Shamir's secret sharing method, which is then embedded into the alpha-channel plane of the cover PNG image. Coefficient parameters involved in the Shamir method are used as *carriers* of the data to be hidden in the proposed method. A prime number used in the method, which is found to dominate the resulting visual quality and data hiding capacity of the stego-image, is properly selected. Also, a mapping function is designed for adjusting the alpha-channel values to create uniform transparency in the alpha-channel plane, resulting in an imperceptible effect in the stego-image. The original *R*, *G*, and *B* channels are untouched so that the original image appearance revealed by the color information of these three channels is kept. Fig. 1 illustrates these core ideas of the proposed method.

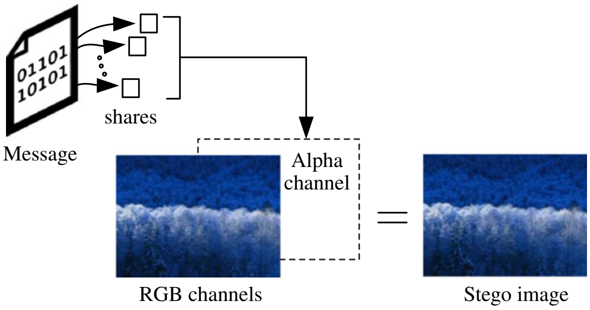


Fig. 1. Illustration of proposed data hiding method via PNG images.

In practical uses, the proposed data hiding method is suitable for applications of *image authentication* and *metadata hiding*. In particular, the application of the proposed method to image authentication is investigated in detail in this study and relevant algorithms are proposed in this paper.

Some other merits of the proposed method are described in the following.

(1) *Causing no destruction to the original cover image* — The proposed method manipulates the alpha-channel plane but leaves the RGB color channels untouched in the entire data hiding and extraction process, resulting in a stego-image with a destruction-free color content, which is beneficial for certain applications like metadata hiding for digital archiving.
(2) *Having the discardability of the alpha channel* — Since data are embedded into the alpha channel in the proposed method, the alpha channel can be discarded for the purpose of obtaining the original image content. In the application of image authentication, an *intact* protected image can be regained after the process of authentication. The proposed method does not leave permanent distortion to an input image as that done by conventional image authentication methods.
(3) *Fully using channels in images for data hiding* — Different from commonly seen color images with three color channels, namely, *R*, *G*, and *B*, the PNG image format has a fourth channel, namely, *alpha*. By the proposed method, the alpha channel of a PNG image can be used to offer an additional data hiding channel for various applications.

The remainder of this paper is organized as follows. In Section 2, the Shamir method on which the proposed data hiding method is based is reviewed first. In Section 3, the details of the proposed method, including the data embedding and extraction processes, are described. In Section 4, algorithms for image authentication as an application of the proposed data hiding method are described. A discussion on security consideration of the proposed method is given in Section 5. Some experimental results are shown in Section 6. Finally, conclusions are made in Section 7.

2. Review of the Shamir method for secret sharing

The proposed method for data hiding is based on the so-called (k, n)-threshold secret sharing method proposed by Shamir [11], where n is the number of participants in the secret sharing activity and k is a threshold specifying the minimum number of shares which should be collected to recover the secret. The detail of the method is reviewed as an algorithm in the following.

Algorithm 1. (k, n)-threshold secret sharing.

Input: a secret d in the form of an integer, the number n of participants, and a threshold k not larger than n.
Output: n shares in the form of integers for the n participants to keep, respectively.

Step 1. Choose a prime number p randomly.

Step 2. Select $k-1$ integer values $c_1, c_2, \ldots, c_{k-1}$ within the range of 0 through $p-1$.

Step 3. Select n distinct real values $x_1, x_2, \ldots, x_n$.

Step 4. Use the following $(k-1)$-degree polynomial to generate n equations to compute n function values $F(x_i)$, called *partial shares*, as follows:

$$F(x_i) = (d + c_1x_i + c_2{x_i}^2 + \cdots + c_{k-1}{x_i}^{k-1})_{\text{mod}p}, \tag{1}$$

where $i=1, 2, \ldots, n$.

Step 5. Deliver the 2-tuple $(x_i, F(x_i))$ as a share to the i th participant, respectively, where $i=1, 2, \ldots, n$.

Since there are k coefficients, including d and c_1 through c_{k-1}, in Eq. (1), it is necessary to collect at least k shares from the n participants to form k equations of the form of (1) to solve these k coefficients. This explains the term, *threshold*, for k as well as the name, (k, n)-*threshold*, for the Shamir method [11]. Below is a description of such a way of equation solving for secret recovery.

Algorithm 2. Secret recovery.

Input: k shares collected from the n participants with k being the threshold mentioned in Algorithm 1.
Output: the secret d hidden in the shares; and the prime number p and the coefficients c_i used in the equations described by (1) in Algorithm 1, where $i=1, 2, \ldots, k-1$.

Steps.

Step 1. Use the k shares $(x_1, F(x_1))$, $(x_2, F(x_2))$, ..., $(x_k, F(x_k))$ to set up the following equations:

$$F(x_j) = (d + c_1x_j + c_2x_j^2 + \cdots + c_{k-1}x_j^{k-1})_{\text{mod}p}, \tag{2}$$

where $j=1, 2, \ldots, k$.

Step 2. Solve the k equations above by Lagrange's interpolation to obtain the desired secret value d [22] as follows:

$$d = (-1)^{k-1}\left[F(x_1)\frac{x_2x_3\cdots x_k}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_k)} + F(x_2)\frac{x_1x_2\cdots x_k}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_k)} + \cdots + F(x_k)\frac{x_1x_2\cdots x_{k-1}}{(x_k-x_1)(x_k-x_2)\cdots(x_k-x_{k-1})}\right]_{\text{mod }p}.$$

Step 3. Compute the values c_1 through c_{k-1} by expanding the following equality and comparing the result with (2) in Step 1 while regarding the variable x in the equality below to be x_j in (2):

$$F(x) = \left[F(x_1)\frac{(x-x_2)(x-x_3)\cdots(x-x_k)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_k)} + F(x_2)\frac{(x-x_1)(x-x_3)\cdots(x-x_k)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_k)} + \cdots + F(x_k)\frac{(x-x_1)(x-x_2)\cdots(x-x_{k-1})}{(x_k-x_1)(x_k-x_2)\cdots(x_k-x_{k-1})}\right]_{\text{mod }p}.$$

Step 3 in the above algorithm is included additionally for the purpose of computing the values of the parameters c_i in the proposed method. In other applications, if only the secret value d need be recovered, this step may be eliminated.

3. Proposed method for data hiding via PNG images

Based on the Shamir method [11] described previously, the basic idea of the proposed method for hiding a given data string M in a cover PNG image I to yield a stego-image I' is described as four major steps as follows.

1. Transform M into a sequence M' of integers.
2. Take sequentially a number of integers from M' as the values of d and c_i in Eq. (1) to compute partial shares $F(x_i)$.
3. Embed $F(x_i)$ into I by replacing some alpha-channel values of I with $F(x_i)$.
4. Repeat (2) and (3) until no integer is left in M', resulting in a stego-image I'.

A block diagram describing the processes in the proposed method is shown in Fig. 2, and the details of the ways we embed and extract the shares are presented as algorithms in the following.

Algorithm 3. Data embedding by secret sharing using a PNG image.

Input: a cover PNG image I and a secret message M in the form of a binary data string.
Output: a stego-image I' in the PNG format.

Steps.

Step 1. (*Initialization*) Divide M into t-bit segments with $t=3$, and transform each segment into an integer, resulting in an integer sequence $M'=d_1d_2d_3\ldots$ where $0 \le d_i \le 7$.

Step 2. (*Beginning of looping*) Take the first four elements from M' as m_1, m_2, m_3, and m_4, starting from the beginning of M'.

Step 3. (*Partial share creation*) Set p, c_i, and x_i in Eq. (1) of Algorithm 1 to be the following values:

(a) $p=11$ (the smallest prime number larger than 7);
(b) $d=m_1$, $c_1=m_2$, $c_2=m_3$, and $c_3=m_4$;
(c) $x_1=1$, $x_2=2$, $x_3=3$, and $x_4=4$,

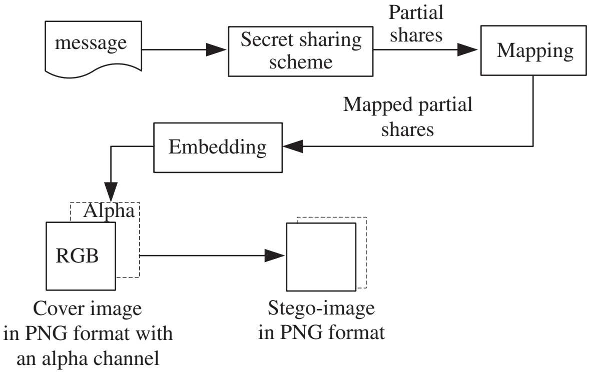


Fig. 2. Block diagram of proposed data hiding processes.

resulting in the following equations:

$$
\begin{aligned}
q_1 &= F(x_1) = (m_1 + m_2x_1 + m_3x_1^2 + m_4x_1^3)_{\mathrm{mod}p},\\
q_2 &= F(x_2) = (m_1 + m_2x_2 + m_3x_2^2 + m_4x_2^3)_{\mathrm{mod}p},\\
q_3 &= F(x_3) = (m_1 + m_2x_3 + m_3x_3^2 + m_4x_3^3)_{\mathrm{mod}p},\\
q_4 &= F(x_4) = (m_1 + m_2x_4 + m_3x_4^2 + m_4x_4^3)_{\mathrm{mod}p};
\end{aligned}
\tag{4}
$$

and compute the partial shares of q_1 through q_4 accordingly.

Step 4. (*Mapping of partial share values*) Add 245 to each of q_1 through q_4 to form q_1', q_2', q_3', and q_4', respectively.

Step 5. (*Data embedding*) Embed q_1' through q_4' into the alpha-channel plane of I in the following way.

5.1. Take in a raster-scan order four unprocessed pixels of I and set their alpha-channel values to be q1′ through q4′, respectively.

5.2. Remove m_i through m_i+3 from M'.

Step 6. (*End of looping*) If M' is not empty, then go to Step 2 to process the next four integers in M'; otherwise, take the final I as the desired stego-image I'.

The above algorithm can be regarded as a (4, 4)-threshold secret sharing method. The possible values of q_1 through q_4 yielded by Eq. (4) in Step 3 of the above algorithm and inserted in the alpha channels of I are between 0 and 10 because the value of p used in Eq. (4) is 11. After performing Step 4 of the algorithm, the values of q_1' through q_4' form a small range of integer values from 245 to 255 which are then embedded into the alpha channels of the cover image I. The distribution of the alpha-channel values within such a small range of values means that very similar values appear everywhere in the alpha channels, resulting in a nearly *uniformly transparent* PNG image, as desired.

Also, it can be seen from the algorithm that every four 3-bit segments of the secret data string are embedded into the alpha-channel values of four pixels of the cover image I to yield the stego-image I'. This means that if the size of the cover image is S, then the data hiding capacity is $R=(4\times3)\times(S/4)=3S$ bits. This is for the case of $t=3$ where t is as mentioned in Step 1. More generally, if every four t-bit segments are transformed and embedded similarly, then it is easy to figure out that $R=(4\times t)\times(S/4)=tS$ bits, which means that the data hiding capacity is proportional to the chosen value of t. Since S is the dimension of the cover image, this capacity of tS is large in general.

However, it should be noted that the larger the value of t is chosen to be, the lower the visual quality of the stego-image will become. The reason is that a larger value of t, according to Step 2 of Algorithm 1, implies that a larger value of p is chosen, and so the possible values of q_1 through q_k, according to Step 3 of Algorithm 3, will be spread in a larger range of values from 0 through $p-1$ due to the use of the mod-p operation. This will cause a wider range of alpha-channel values even after the value mapping of Step 4 of Algorithm 3 is conducted. This wider alpha-channel value range in turn leads to a more obvious non-uniform transparency effect appearing on the stego-image. This also explains the reason why we segment, in Step 1 of Algorithm 3, the message M into segments of $t=3$ bits for use in Eq. (4), which is a compromise between the resulting data hiding capacity and stego-image quality according to our experimental experience. Of course, if a higher image quality of the stego-image is required, we may use a smaller t like $t=2$ at the sacrifice of the data hiding capacity.

In addition, it is noted that Algorithm 3 takes every *four* integer numbers of the string M' each time and embeds them into the alpha channels of *four* pixels of the cover image. It is not difficult to figure out that the algorithm can be *generalized* to take n integers each time and embeds them into n pixels. For this, just modify part of Step 3 to be as follows:

…

(b) $d=m_1, c_1=m_2, c_2=m_3, \ldots, c_n=m_n$;

(c) $x_1=1, x_2=2, x_3=3, \ldots, x_n=n$,

resulting in the following equations:

$$
\begin{aligned}
q_1 &= F(x_1) = (m_1 + m_2x_1 + m_3x_1^2 + \cdots + m_nx_1^n)_{\mathrm{mod}p},\\
q_2 &= F(x_2) = (m_1 + m_2x_2 + m_3x_2^2 + \cdots + m_nx_2^n)_{\mathrm{mod}\ p},\\
q_3 &= F(x_3) = (m_1 + m_2x_3 + m_3{x_3}^2 + \cdots + m_nx_3^n)_{\mathrm{mod}\ p},\\
&\vdots\\
q_n &= F(x_n) = (m_1 + m_2x_n + m_3x_n^2 + \cdots + m_nx_n^n)_{\mathrm{mod}\ p}.
\end{aligned}
\tag{4′}
$$

The resulting generalized algorithm yields, as can be figured out again, a data hiding capacity of $R=(n\times t)\times(S/n)=tS$ bits which is identical to the original algorithm.

Now, the process of hidden data extraction is described in the following. A block diagram describing the proposed data extraction processes is shown in Fig. 3.

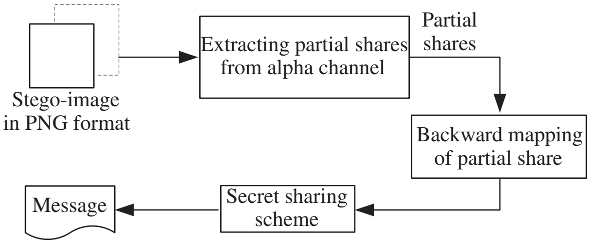

Stego-image in PNG format
Extracting partial shares from alpha channel
Partial shares
Backward mapping of partial share
Secret sharing scheme
Message

Fig. 3. Block diagram of proposed data extraction processes.

Algorithm 4. Data extraction from a stego-image.

Input: a stego-image I' created by Algorithm 3 in the PNG format.
Output: the binary data string M hidden in I'.

Steps.

```
Step 1. (Initialization) Create an empty string $M$.
Step 2. (Beginning of looping) Take in a raster-scan order four alpha-channel values $q_1'$, $q_2'$, $q_3'$, and $q_4'$ from $I'$.
Step 3. (Backward mapping of partial share values) Subtract 245 from each of $q_1'$ through $q_4'$ to obtain $q_1$ through $q_4$, respectively.
Step 4. (Data extraction) Perform the secret recovery process described by Algorithm 2 to extract four integers $m_1$ through $m_4$ hidden in $q_1$ through $q_4$, transform $m_1$ through $m_4$ into binary numbers, and append them in a sequential order to the end of $M$.
Step 5. (End of looping) If all shares embedded in $I'$ are processed, then take the final $M$ as output; otherwise, go to Step 2.
```

Similarly to generalization of Algorithm 3 as mentioned previously, Algorithm 4 above may also be generalized to take care of data extraction from images yielded by the generalized version of Algorithm 3. The details are simple and so omitted.

4. Application of proposed method to image authentication

A possible application of data hiding is *image authentication*. To implement image authentication by the proposed data hiding method, an input image with RGB channels (like a BMP image) to be protected is transformed first into a PNG image with the alpha-channel values all set initially to be 0. The PNG image is processed next to generate color-dependent authentication signals, which are then embedded into the alpha channel by using the previously proposed data hiding method to yield the resulting stego-image. Then, in the later authentication signal verification process, the embedded authentication signals are extracted from the alpha channel plane of a stego-image and verified against those computed from the color values of the pixels of the image. A pixel with mismatched signals is regarded as being tampered with. Fig. 4 illustrates the processes of authentication signals generation and embedding in the proposed color image authentication method, and Fig. 5 illustrates the image verification processes of the proposed method. More details are described as two algorithms (Algorithms 5 and 6) in the following.

Algorithm 5. Authentication signal generation and embedding.

Input: a PNG image I and a key K.
Output: a protected image I' with authentication signals embedded.

Steps.

```
Step 1. (Initialization and beginning of looping) Take in a raster-scan order three pixels $p_1$, $p_2$, and $p_3$ from $I$, and use $K$ as the seed for a random number generator to get a number sequence $S=s_1, s_2, \ldots$.
Step 2. (Creation of authentication signals) For each of $p_1$, $p_2$ and $p_3$, denoted as $p_i$, perform the following steps.
    2.1 Take the $R$, $G$, and $B$ values of $p_i$, denoted as $V_R$, $V_G$, and $V_B$, respectively.
    2.2 Take in order three elements from $S$, denoted as $s_j$, $s_{j+1}$, and $s_{j+2}$, to perform the following operations to obtain three bits $r$, $g$, and $b$, respectively:
        $(V_R+s_j)_{\text{mod}2}=r$;
        $(V_G+s_{j+1})_{\text{mod}2}=g$;
        $(V_B+s_{j+2})_{\text{mod}2}=b$.
    2.3 Concatenate $r$, $g$, and $b$ as a 3-bit string $rgb$, and transform it into a decimal integer as the authentication signal $D_i$ for $p_i$.
Step 3. (Authentication signal embedding into three pixels) Perform the previously-described generalized
```

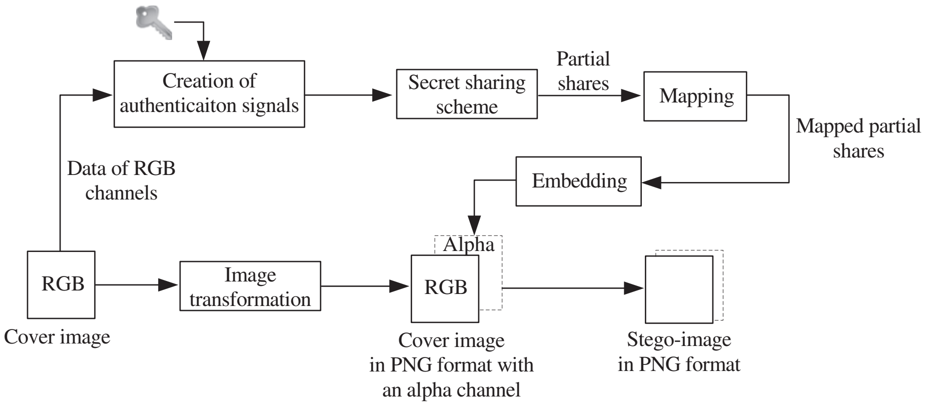


Fig. 4. Block diagram of generating a stego-image with authentication signals.

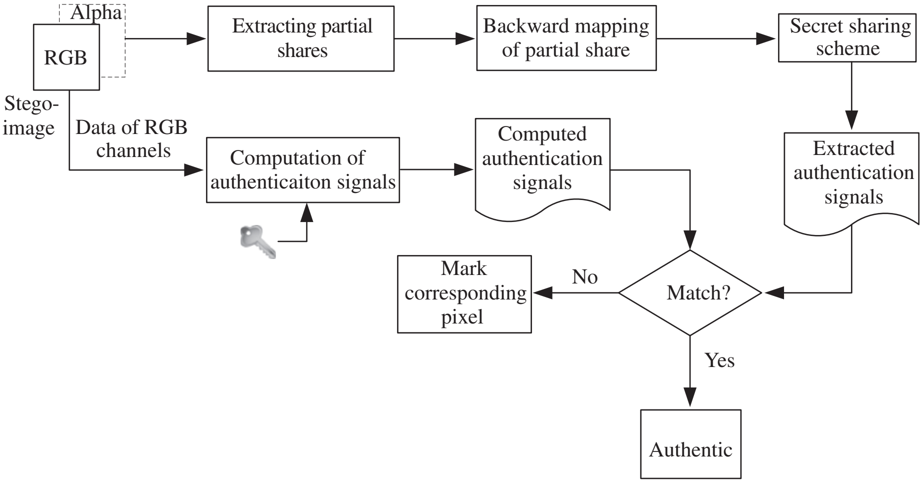

Alpha
RGB
Stego-image
Extracting partial shares
Backward mapping of partial share
Secret sharing scheme
Data of RGB channels
Computation of authenticaiton signals
Computed authentication signals
Extracted authentication signals
Mark corresponding pixel
No
Match?
Yes
Authentic

Fig. 5. Block diagram of verifying a stego-image.

version of Algorithm 3 to embed the three authentication signals D_1, D_2, and D_3 into three pixels of I, except that in the step of *mapping of partial share values* (Step 4) in Algorithm 3, 244 instead of 245 is added to each partial share value.

Step 4. (*End of looping*) If there exists any unprocessed pixel in I, then go to Step 2; otherwise, take the final I as the desired stego-image I'.

Algorithm 6. Authentication signal verification.

Input: a protected stego-image I' created by Algorithm 5, and a key K used there.
Output: image I' with altered pixels being marked if found.

Steps.

Step 1. (*Initialization and beginning of looping*) Take in a raster-scan order three pixels p_1, p_2 and p_3 from I'; let their alpha-channel values be denoted as q_1', q_2', and q_3', respectively; and use K as the seed for a random number generator to generate a random number sequence $S=s_1, s_2, \ldots$.

Step 2. (*Extraction of authentication signals embedded in alpha channels*) Perform the previously-described generalized version of Algorithm 4 with q_1', q_2', and q_3' as input to extract the authentication signals D_1', D_2', and D_3'cept that in the step of *backward mapping of partial share values* (Step 3) in Algorithm 4, 244 instead of 245 is subtracted from each of q_1' through q_3'.

Step 3. (*Computation of authentication signals from pixels' color values*) For each of p_1, p_2 and p_3, denoted as p_i, perform the following steps.

2.1. Take the R, G, and B values of p_i, denoted as V_R, V_G, and V_B, respectively.

2.2. Take in order three elements from S, denoted as s_j, s_{j+1}, and s_{j+2}, to perform the following operations to obtain three bits r, g, and b, respectively:

$$(V_R+s_j)_{\text{mod}2}=r;$$
$$(V_G+s_{j+1})_{\text{mod}2}=g;$$
$$(V_B+s_{j+2})_{\text{mod}2}=b.$$

2.3. Concatenate r, g, and b as a 3-bit string rgb, and transform it into a decimal integer as the authentication signal D_i for p_i.

Step 4. (*Matching of extracted and computed authentication signals*) Match D_1, D_2, and D_3 with D_1', D_2', and D_3', respectively, and if there exists any mismatched pair, then mark the corresponding pixel as being tampered with in the input image I'.

Step 5. (*End of looping*) If there exists any unprocessed pixel in I', then go to Step 2; otherwise, take the final I', possibly marked, as output.

Totally six algorithms so far having been introduced in this study, Table 1 is therefore given for a rapid reviewing of the previously-described algorithms' content.

5. Security consideration

The secret key K used in Step 2.2 of Algorithm 5 provides a measure to protect the authentication signals to be counterfeited. More specifically, since three bits consist of an authentication signal for each pixel, the probability of correctly guessing an authentication signal for each pixel is 1/8. To enhance further the security of the proposed method, an additional measure is adopted but not included in the above algorithms for clarity of algorithm descriptions. It is *randomization* of the constant values of x_1 through x_n used in Step 3 of the generalized version of Algorithm 3 within the allowed integer range of $0 \le x_i < p$ [11] with the help of another secret key. Then, the probability of correctly guessing values of x_1 through x_3 used in a set of three pixels can be

Table 1
Six algorithms described in this study and the corresponding descriptions.

Algorithms	Descriptions
Algorithm 1	Encoding phase of (k, n)-threshold secret sharing method
Algorithm 2	Decoding phase of (k, n)-threshold secret sharing method
Algorithm 3	Proposed data embedding method based on Algorithm 1
Algorithm 4	Proposed data extraction processes based on Algorithm 2
Algorithm 5	Proposed color image authentication method based on Algorithm 3
Algorithm 6	Proposed color image verification processes based on Algorithm 4

figured out to be

$$1/[p\times(p-1)\times\cdots\times(p-n+1)]=1/[11\times(11-1)\times(11-3+1))]=1/990\approx 0.1\%.$$

As a result, the possibility of correctly guessing all these values used in a stego-image of size S is $1/[p\times(p-1)\times\cdots\times(p-n+1)]^{S/n}$ which is small enough in general cases. Through the use of above security measures, it is almost impossible for an attacked color stego-image with counterfeit authentication signals to pass the proposed image authentication processes.

Furthermore, the authentication data may possibly be erased by discarding or modifying the content of the alpha channel without producing any alteration to the content of the color channels. For this, it is noted that the aim of the image authentication is to make a stego-image be trusted by an authentication side after the authentication process. Accordingly, avoiding a counterfeit image to pass the authentication process is a major concern, which is different from the major concern of robustness in the copyright protection — a surviving and recognizable watermark such as a logo has to be extracted from the stego-image even after attacks. Therefore, at an authentication side, if a stego-image ready to be authenticated has no the alpha channel, it means the integrity of the stego-image is seriously lost because an undamaged stego-image must have the alpha channel. In short, if an attacker erases the authentication data by just discarding or modifying the alpha channel of the stego-image and producing no alteration to the image color channels, such a kind of attack will evidently be found by the proposed method and the malicious aim to deceive the authentication processes will fail.

6. Experimental results

In this section, experimental results of the proposed data hiding method and the comparison with existing methods in data hiding are described first. Subsequently, experimental results of the proposed color image authentication method, which is based on the aforementioned proposed data hiding method, and the comparison with existing methods in color image authentication are given.

6.1. Experimental results of data hiding using color images

A lot of experiments have been conducted to test the proposed algorithms on totally above 100 color images. Some results using test images, named Lena, jet, Tiffany, Baboon, boat and mountain as given in Fig. 6(a), (c), (e), (g), (i) and (k), respectively, are shown here. The results of applying the proposed method using Algorithm 3 to embed a long sequence of binary message data into the three images are shown in Fig. 6(b), (d), (f), (h), (j) and (l), respectively. As can be seen from the figures, the stego-images are visually almost identical to the cover images, respectively, although the alpha-channel contents of the stego-images include embedded message data. Note that in Algorithm 3, the value of t, which is the number of bits taken as a segment and transformed into an integer for use in the secret sharing process performed by Algorithm 3, was taken to be 3.

Next, we show the effect of choosing different values of t in Step 1 of Algorithm 3. Recall that the data hiding capacity has been computed to be $R=tS$ where S is the size of the cover image. Although accordingly t may be chosen to be the maximum of 7 to increase the data hiding capacity, the yielded stego-image quality degrades very much, as illustrated by the results shown in Fig. 7, where Fig. 7(a) is the stego-image of Lena yielded by Algorithm 3 with t taken to be 7 which includes, as can be seen, a lot of white noise. For comparison, a better result of using $t=3$ shown in Fig. 6(b) is repeated in Fig. 7(b) in which the white noise created by the algorithm is almost imperceptible, as mentioned previously.

Furthermore, we show in more detail the data hiding capacities and the corresponding stego-image qualities for all possible values of $t=1, 2, \ldots, 7$ for the aforementioned six images Lena, Jet, Tiffany, Baboon, Pepper and Mountain used as cover images. Specifically, in Tables 2 and 3 we show, in addition to the data hiding capacities, the qualities of the alpha-channel planes of the six corresponding stego-images in terms of the MAE (Mean Average Error) measure. The MAE introduced in [23] is an objective metric to measure the visual quality, and the larger value of MAE means the image is with poorer quality. According to the statistics of conducted experiments shown in Tables 2 and 3, the alpha channel provides satisfactory transparency effect when the value of MAE not larger than around 4.25. Furthermore, in Fig. 8, we show the stego-images corresponding to $t=1, 2, \ldots, 7$. As can be seen, when $t=4$, the value of MAE is 4.25 and the corresponding stego-image quality shown in Fig. 8(c) is still visually good. According to the statistics of other conducted experiments shown in Tables 2 and 3, the alpha channel provides sufficient transparency effect when the value of MAE not larger than around 4.25. In addition, the corresponding data hiding capacity is $R=tS=3\times 512\times 512=75\text{ K}=786\,432$ bits, which is good enough for applications.

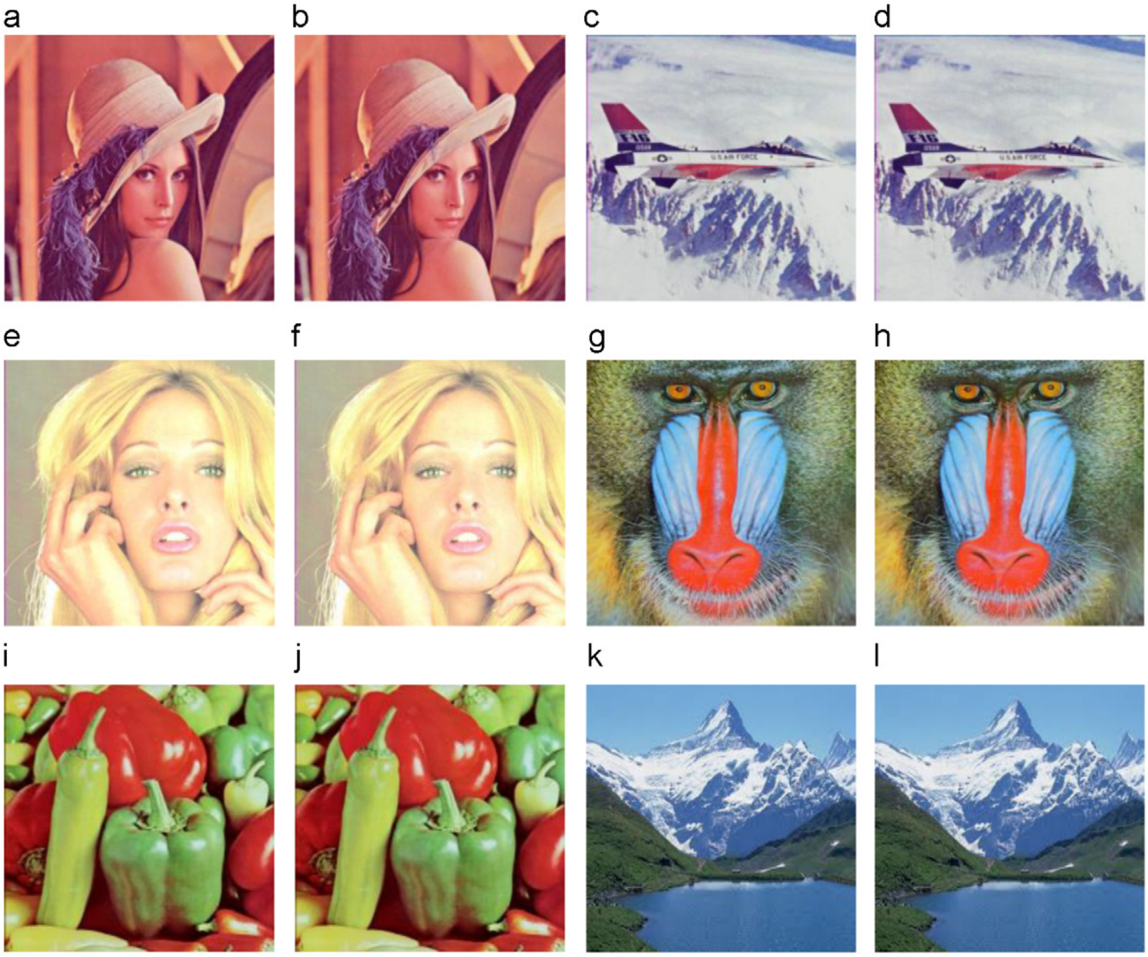


Fig. 6. Results of applying Algorithm 3 to embed a long sequence of binary message data into test images: (a) Cover image Lena; (b) Stego-image of Lena; (c) Cover image jet; (d) Stego-image of jet; (e) Cover image Tiffany; (f) Stego-image of Tiffany; (g) Cover image baboon; (h) Stego-image of baboon; (i) Cover image pepper; (j) Stego-image of pepper; (k) Cover image mountain; (l) Stego-image of mountain.

In fact, it can be seen from Fig. 8(d) that even when t is taken to be 4, the stego-image quality is still acceptable with the data hiding capacity increased to 100 K bits. On the contrary, if image quality is the most serious concern, then t may be reduced to 2 or even 1 at the sacrifice of the data hiding capacity. Note that the data hiding capacity is related to the t value only; it is independent of the cover image content. It is also noted that, different from other methods, the channels of R, G, and B of the cover image are not processed by the proposed method, yielding a lossless result in the color channels.

To further demonstrate the feasibility of the proposed method for metadata embedding, an art image of Gleaner shown in Fig. 9(a) and the history data related to this art image shown in Fig. 9(b) are respectively taken to be the input cover image and the metadata. Since the size of the metadata is 1.85 KB (=15 156 bits), it is adequate to set the value of t used in Algorithm 3 to be 1 for embedding the metadata into Fig. 9(a). The resulting stego-image with the MAE value of 0.11 is shown in Fig. 9(c) and, as can be seen, the stego-image is visually identical to the cover image of Fig. 9(a). Finally, the result of applying the proposed Algorithm 4 to extract the embedded metadata from Fig. 9(c) is shown in Fig. 9(d).

6.2. Comparison with existing data hiding method

To measure the performance of the proposed method, we compare the proposed method with Kim's method [26] which produces high data hiding capacity with the good visual quality of a stego-image in the field of lossless data hiding. Importantly, both the proposed method and Kim's method have a parameter t that can be adjusted to control the data hiding capacity. With the increasing of the value of t, the data hiding capacity increases while the quality of the stego-image decreases for both methods. In more detail, in Kim's method [26], the parameter t denotes the number of hiding levels; in the proposed method, the t denotes the number of bits of a message segment as mentioned in Step 1 of Algorithm 3. As can be observed from Table 4, the proposed method gets a significant improvement in hiding capacity, while causes no destruction to the original content of RGB channels.

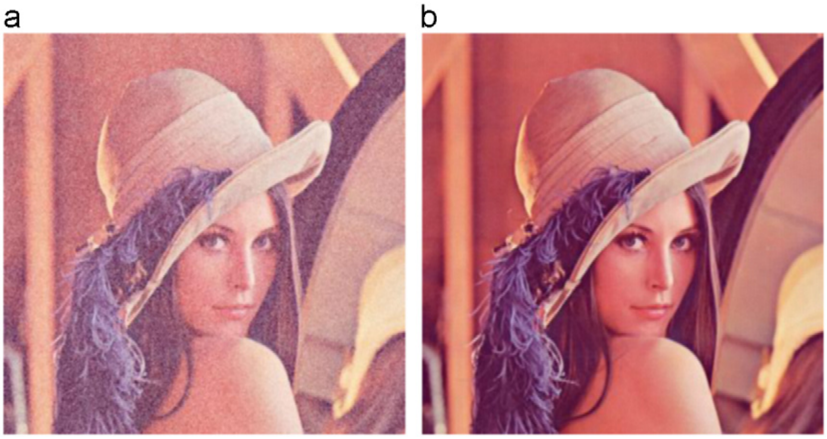


Fig. 7. An example of experimental results showing compromise between data hiding capacity and stego-image quality: (a) Stego-image of Lena yielded by choosing $t=7$ with more data hidden but showing more white noise coming from the alpha channel; (b) Stego-image of Lena yielded by choosing $t=3$ with less data hidden but with almost no noise due to nearly total transparency in the alpha channel.

To further reveal the performance of the proposed method, we also compare the proposed method with Luo's method [27]. The method of [27] uses interpolation technique to fulfill the reversible watermarking and outperforms many existing methods such as [28–29]. As can be seen from Table 5, the proposed method yields more data hiding capacity than that of the method of [27].

6.3. *Experimental results of color image authentication*

In this section, we show some experimental results of applying the proposed image authentication algorithms (Algorithms 5 and 6) to authenticate stego-images attacked by two common image editing operations, i.e., superimposing and painting. It was found that if the superimposing operation is used in the attack, the alpha channel values will be replaced with the new value 255 at the attacked part. Since the largest alpha channel value generated by the proposed method is 254 (see Step 3 in Algorithm 5), attacked pixels can be easily detected and marked by checking the existence of the specific value 255 in the alpha channel. As an example, Fig. 10(a) shows a cover image "Tiffany" and Fig. 10(b) is the stego-image of Tiffany. In Fig. 10(c), the stego-image was attacked by superimposing a fake mouth on the face. Fig. 10(d) shows the authentication result in which all altered pixels were detected and marked in black. Table 6 includes the statistics of the performance of the proposed method shown by the above experimental results in terms of the three parameters: *detection ratio*, *false acceptance ratio*, and *false rejection ratio*, which are defined in the following:

(1) detection ratio=(the number of detected pixels)/(the number of tampered pixels);
(2) false acceptance ratio=(the number of tampered pixels marked as untampered)/(the total number of tampered pixels);
(3) false rejection ratio=(the number of untampered pixels marked as tampered)/(the total number of untampered pixels).

As can be observed from Table 6, both the false acceptance ratio and false rejection ratio are 0% for the reason that, as mentioned previously, alpha channel values 255 only occur at attacked pixels.

Another example of a tampered image attacked by superimposing a rose to the hair part of Tiffany is shown in Fig. 11(b). The authentication result is shown in Fig. 11(c) in which all pixels consisting of the added rose were successfully detected and marked in black. The corresponding statistics is also given in Table 6.

Some experimental results of using painting operations to attack stego-images are shown in Fig. 12. Specifically, Fig. 12(a) and (b) are respectively an input cover image "jet" and a generated stego-image. In Fig. 12(c), a logo and words printed on the body of the plane were smeared by painting color similar to the plane body. The authentication result generated from Algorithm 6 is shown in Fig. 12(d) in which tampered regions were successfully detected and marked in black. However, as can be seen, some tampered pixels were not detected and appeared as noise in the marked region. This phenomenon results from the case that the authentication signals extracted from the alpha channel incidentally match the authentication signals computed from the tampered pixels. This is also the reason why the false acceptance ratio exists. It is noted that the alpha channel content keeps intact after the painting operation and so the extracted authentication signals are always true, yielding the false rejection ratio is 0%. Related statistics of the experiment is given in Table 7.

Since the authentication signal of each pixel is composed of three bits, there is a probability of 1/8 for an erroneous authentication, leading to a false acceptance ratio of around 12.5%. As an example, Fig. 13(b) shows that the stego-image of jet was tampered with by smearing the entire plane out of the image content. The authentication result is shown in Fig. 13(c) in which

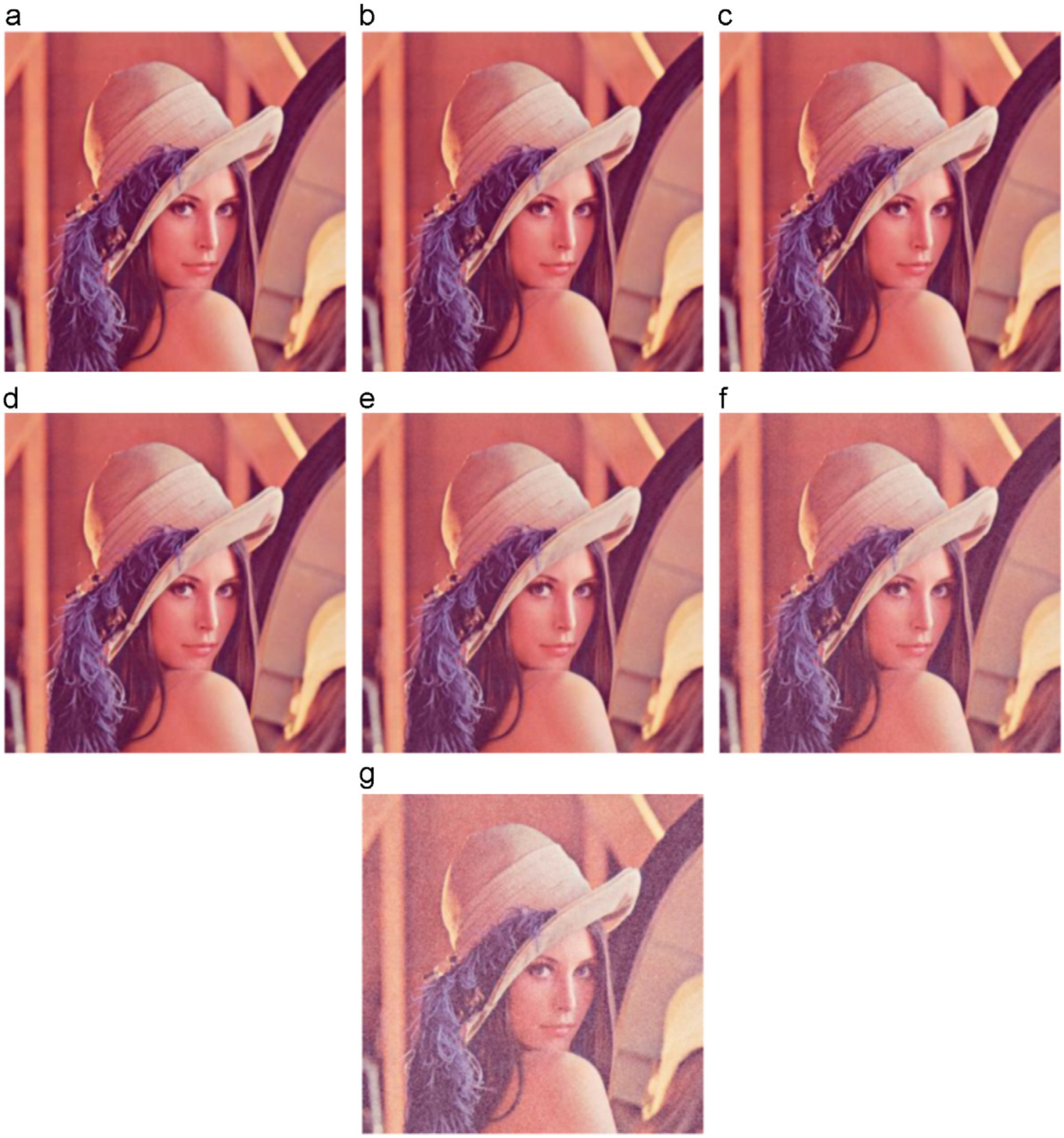


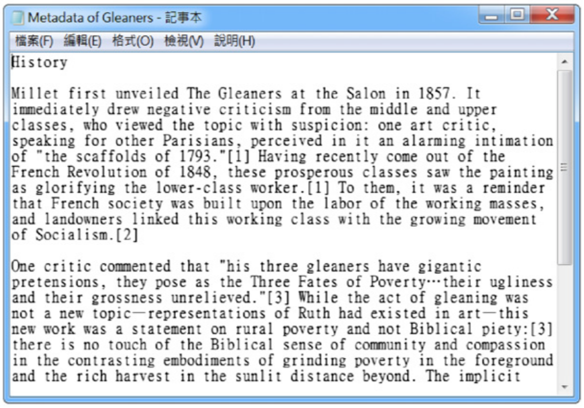
Fig. 8. Stego-images yielded by Algorithm 3 for $t=1$–7: (a)–(g) correspond to $t=1, 2, \ldots, 7$, respectively.

Table 2
Data hiding capacities and stego-image qualities for $t=1$–7 using test images Lena, jet and Tiffany (DHC=data hiding capacity; MAE=mean average error; PSNR=peak of signal to noise ratio).

t value	Lena			Jet			Tiffany		
	DHC (bits)	MAE of alpha channel (dB)	PSNR of RGB channels (dB)	DHC (bits)	MAE of alpha channel (dB)	PSNR of RGB channels (dB)	DHC (bits)	MAE of alpha channel (dB)	PSNR of RGB channels (dB)
$t=1$	262 144	0.720	∞	262 144	0.721	∞	262 144	0.721	∞
$t=2$	524 288	1.210	∞	524 288	1.203	∞	524 288	1.211	∞
$t=3$	786 432	2.732	∞	786 432	2.737	∞	786 432	2.733	∞
$t=4$	1 048 576	4.250	∞	1 048 576	4.233	∞	1 048 576	4.251	∞
$t=5$	1 310 720	9.246	∞	1 310 720	9.238	∞	1 310 720	9.246	∞
$t=6$	1 572 864	16.236	∞	1 572 864	16.242	∞	1 572 864	16.238	∞
$t=7$	1 835 008	32.837	∞	1 835 008	32.835	∞	1 835 008	32.833	∞

Table 3
Data hiding capacities and stego-image qualities for $t=1$–7 using test images Baboon, Boat and Mountain (DHC=data hiding capacity; MAE=mean average error; PSNR=peak of signal to noise ratio).

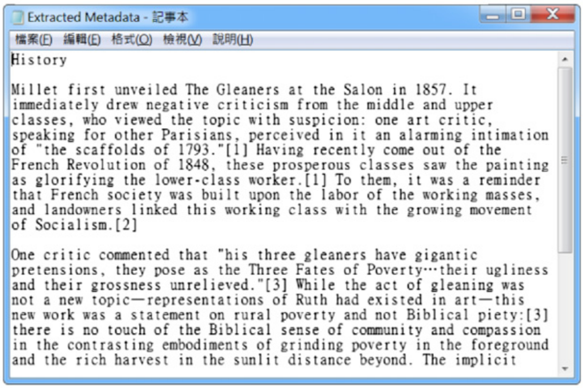
t value	Baboon			Boat			Mountain		
	DHC (bits)	MAE of alpha channel (dB)	PSNR of RGB channels (dB)	DHC (bits)	MAE of alpha channel (dB)	PSNR of RGB channels (dB)	DHC (bits)	MAE of alpha channel (dB)	PSNR of RGB channels (dB)
$t=1$	262144	0.723	∞	262144	0.719	∞	262144	0.722	∞
$t=2$	524288	1.215	∞	524288	1.213	∞	524288	1.220	∞
$t=3$	786432	2.732	∞	786432	2.727	∞	786432	2.722	∞
$t=4$	1048 576	4.246	∞	1048 576	4.220	∞	1048 576	4.233	∞
$t=5$	1310 720	9.264	∞	1310 720	9.262	∞	1310 720	9.259	∞
$t=6$	1572 864	16.219	∞	1572 864	16.231	∞	1572 864	16.222	∞
$t=7$	1835 008	32.839	∞	1835 008	32.823	∞	1835 008	32.840	∞


b
Metadata of Gleaners - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

History

Millet first unveiled The Gleaners at the Salon in 1857. It immediately drew negative criticism from the middle and upper classes, who viewed the topic with suspicion: one art critic, speaking for other Parisians, perceived in it an alarming intimation of "the scaffolds of 1793."[1] Having recently come out of the French Revolution of 1848, these prosperous classes saw the painting as glorifying the lower-class worker.[1] To them, it was a reminder that French society was built upon the labor of the working masses, and landowners linked this working class with the growing movement of Socialism.[2]

One critic commented that "his three gleaners have gigantic pretensions, they pose as the Three Fates of Poverty…their ugliness and their grossness unrelieved."[3] While the act of gleaning was not a new topic—representations of Ruth had existed in art—this new work was a statement on rural poverty and not Biblical piety:[3] there is no touch of the Biblical sense of community and compassion in the contrasting embodiments of grinding poverty in the foreground and the rich harvest in the sunlit distance beyond. The implicit


d
Extracted Metadata - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

History

Millet first unveiled The Gleaners at the Salon in 1857. It immediately drew negative criticism from the middle and upper classes, who viewed the topic with suspicion: one art critic, speaking for other Parisians, perceived in it an alarming intimation of "the scaffolds of 1793."[1] Having recently come out of the French Revolution of 1848, these prosperous classes saw the painting as glorifying the lower-class worker.[1] To them, it was a reminder that French society was built upon the labor of the working masses, and landowners linked this working class with the growing movement of Socialism.[2]

One critic commented that "his three gleaners have gigantic pretensions, they pose as the Three Fates of Poverty…their ugliness and their grossness unrelieved."[3] While the act of gleaning was not a new topic—representations of Ruth had existed in art—this new work was a statement on rural poverty and not Biblical piety:[3] there is no touch of the Biblical sense of community and compassion in the contrasting embodiments of grinding poverty in the foreground and the rich harvest in the sunlit distance beyond. The implicit

Fig. 9. Experimental results of metadata embedding and extraction by applying proposed Algorithms 3 and 4: (a) Cover image Gleaner; (b) History data used as metadata; (c) Stego-image of Gleaner; (d) Metadata extracted from (c).

85.02% of tampered pixels were detected and marked in black. In other words, 14.98% of tampered pixels incidentally passed the authentication process, which meets the probabilistic expectation of around 12.5% authentication misses. The corresponding statistics is also given in Table 7.

In addition, it is noted that though we use the previously mentioned alpha value 255 as a distinguishing one to detect tampering caused by superimposing for the reason of efficiency, we may actually just use Algorithm 6 proposed in this study to deal with such cases normally. The reason is that the proposed method detects attacks by matching authentication signals computed from the color channels and those extracted from the alpha channel. If there exists any *mismatche*d pair, then the corresponding pixel will be marked as being tampered with.

6.4. *Comparison with existing color image authentication methods*

Several existing authentication methods related to color images are taken to compare with the proposed

Table 4
Comparison of performance of proposed method with the method of [26].

(# of levels of [26]) (*t* value of our method)		1	2	3	4	5	6	7
Lena (bits) (dB)	Kim [26]	31 865 (48.98)	81 511 (44.40)	114 536 (42.00)	137 326 (40.47)	153 558 (39.37)	165 641 (38.53)	174 964 (37.86)
	Proposed method	262 144 (∞)	524 288 (∞)	786 432 (∞)	1 048 576 (∞)	1 310 720 (∞)	1 572 864 (∞)	1 835 008 (∞)
Airplane (bits) (dB)	Kim [26]	48 774 (49.17)	104 898 (45.06)	135 903 (42.84)	154 882 (41.34)	167 828 (40.22)	176 870 (39.32)	183 720 (38.57)
	Proposed method	262 144 (∞)	524 288 (∞)	786 432 (∞)	1 048 576 (∞)	1 310 720 (∞)	1 572 864 (∞)	1 835 008 (∞)
Baboon (bits) (dB)	Kim [26]	7249 (48.72)	21 141 (43.02)	34 592 (39.80)	47 424 (37.59)	59 459 (35.95)	70 783 (34.66)	81 216 (33.62)
	Proposed method	262 144 (∞)	524 288 (∞)	786 432 (∞)	1 048 576 (∞)	1 310 720 (∞)	1 572 864 (∞)	1 835 008 (∞)
Boat (bits) (dB)	Kim [26]	21 442 (48.87)	60 903 (43.89)	93 559 (41.32)	117 827 (39.67)	135 222 (38.46)	147 692 (37.49)	156 937 (36.66)
	Proposed method	262 144 (∞)	524 288 (∞)	786 432 (∞)	1 048 576 (∞)	1 310 720 (∞)	1 572 864 (∞)	1 835 008 (∞)

Table 5
Comparison of performance of proposed method with the method of [27].

Methods	Lena		Baboon		Airplane		Boat	
	Capacity	PSNR of color content	Capacity	PSNR of color content	Capacity	PSNR of color content	Capacity	PSNR of color content
Luo et al. [27]	71674	48.82	22696	48.36	84050	48.94	38734	48.50
Proposed ($t=1$)	262144	∞	262144	∞	262144	∞	262144	∞

method for the purpose of presenting contributions made in this study. A comparison table in terms of important capabilities is given in Table 8. As can be observed, since the proposed method makes use of the alpha channel instead of conventional illumination [1] or RGB [24–25] channels for embedding authentication signals, the proposed method is the only one which has the characteristic of losslessness of the image content.

Furthermore, it is a challenge to create the authentication signal for each color pixel and to find embeddable space that is large enough to accommodate these authentication signals. Therefore, though some image authentication methods developed for color images, to the best of our knowledge, only the proposed method and the method of [25] belong to the category of pixel-level authentication methods. This means that other color image authentication methods localize tampered area in the unit of block composed of many pixels, while the proposed method and the method of [25] localize the tampered area in the unit of pixel, yielding a finer localization result. For a fair comparison, we compare the proposed method with the method [25] also developed at the pixel-level authentication precision in terms of the false rejection ratio and false acceptance ratio. As can be seen from Table 9, since the authentication signal of each pixel is composed of three bits in the proposed method, a probability for an erroneous authentication is 1/8, leading to a false acceptance ratio of around 12.5%. However, the authentication signal of each pixel is one bit in the method of [25], so a probability for an erroneous authentication is 1/2, leading to a false acceptance ratio of around 50%. It is noted that, in the method of [25], the authentication signals will not be damaged as long as a protected image without being tampered with, leading to a false rejection ratio of 0%. In addition, as can be seen from Table 8, the localization capability only works when the alteration occurs in the blue channel in [25]. This fact results from that the method of [25] uses the blue channel for embedding authentication signals generated from channels of red and green. As a result, when the image content in the red and green channels altered, the method of [25] is able to detect the existence of the attack but unable to localize alterations precisely.

Moreover, it is worth to note that the method of [1] embeds watermarks in the illumination channel and leaves chrominance channel untouched. Therefore, modifications involved in illumination can be detected efficiently but those made in the chrominance channel will be neglected. In contrast, the proposed method, which generates authentication signals from data of the RGB channels, provides high sensitivity to alterations occurring in each channel.

At last, the method in [1] allows reasonable image processing such as JPEG compression without raising false positive alarms and is practical for some applications. In this aspect, the proposed method yields a stego-image in the PNG format which in normal cases will not be compressed further, reducing the possibility of erroneous authentication caused by incidental image compression.

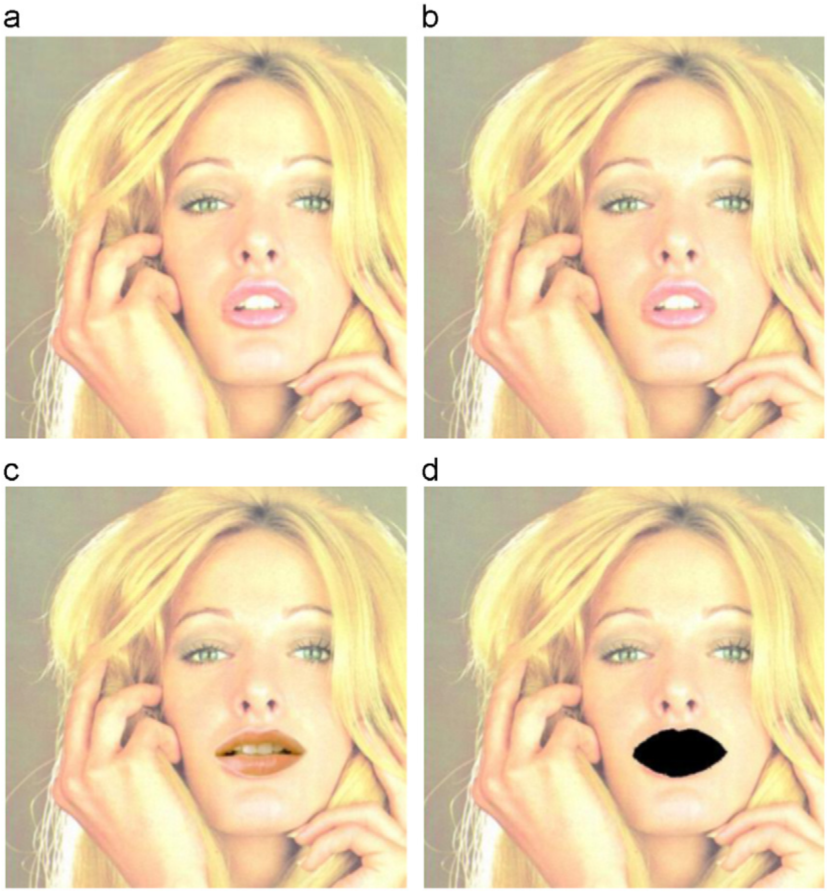


Fig. 10. Authentication result of an attacked stego-image of Tiffany with a pasted fake mouth: (a) Cover image; (b) Stego-image with authentication signals; (c) Modified stego-image; (d) Result with altered pixels detected and marked in black.

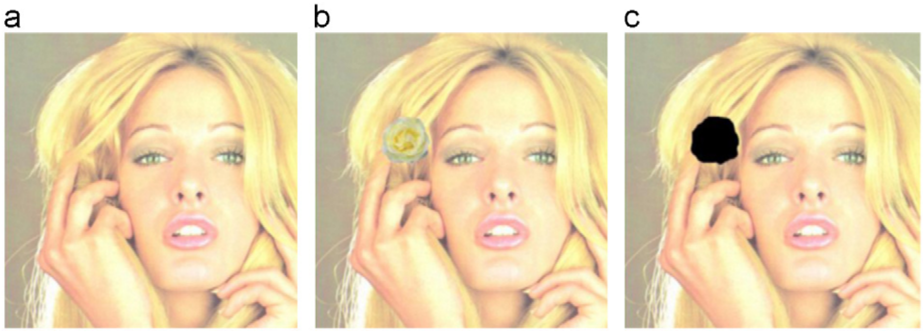


Fig. 11. Authentication result of an attacked stego-image of Tiffany with an added rose: (a) Stego-image with authentication signals; (b) Modified stego-image; (c) Result with altered pixels detected and marked in black.

7. Conclusions

A new type of data hiding via PNG images based on information sharing has been proposed. The Shamir's secret sharing method is used first in a novel way to generate partial shares from a given data string. The alpha-channel plane of a cover PNG image is utilized next to embed the partial shares, yielding a stego-image with undesirable white noise. The white noise is then eliminated skillfully by choosing a small prime number, dividing the input data string into 3-bit segments, and mapping computed share values into a range of large alpha-channel values which create high transparency. Generalization of the method to allow compromise between the resulting data hiding capacity and stego-image quality has also been proposed.

Moreover, detailed algorithms for applying the proposed method to color image authentication have been proposed. Shown by the results is the applicability of the proposed authentication algorithms to detect attacks implemented by common image-content alternation operations. Good experimental results prove the effectiveness of the proposed methods in the aspects of tampering detection ratio, characteristic of losslessness,

Table 6
Statistics of experimental results.

Experimental result (image size=512 × 512)	No. of tampered pixels	No. of detected pixels (detection ratio)	False acceptance ratio (%)	False rejection ratio (%)
Exp. 1 shown in Fig. 9	5886	5886 (100%)	0	0
Exp. 2 shown in Fig. 10	5207	5207 (100%)	0	0

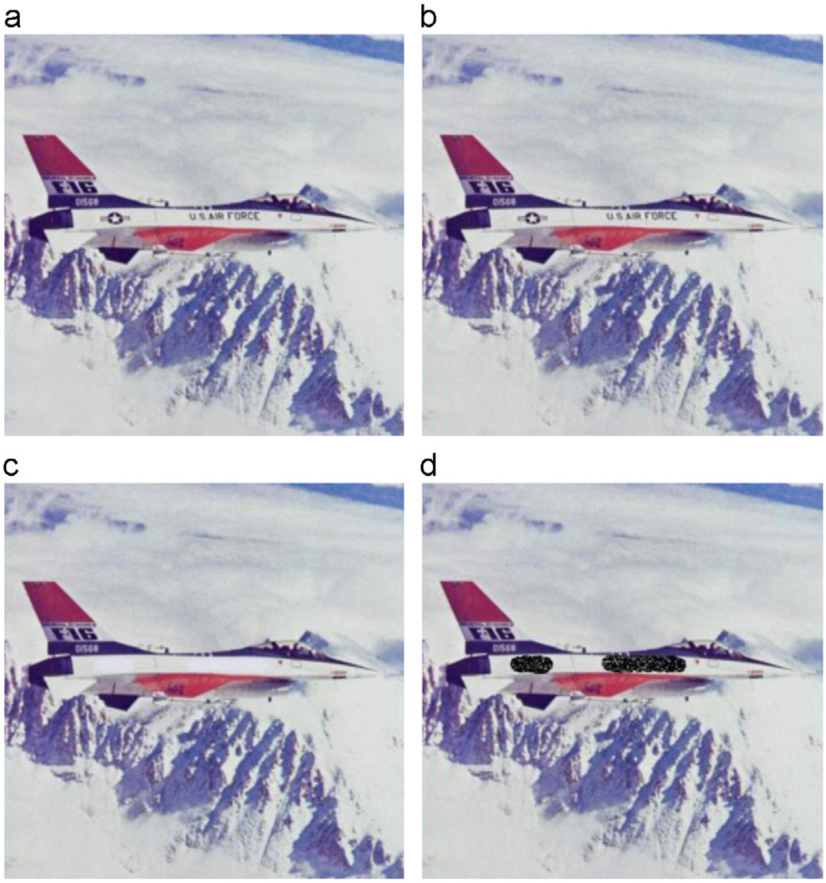


Fig. 12. Authentication result of an attacked stego-image of jet with the smeared logo and words: (a) Stego-image with authentication signals; (b) Modified stego-image; (c) Result with altered pixels detected and marked in black.

Table 7
Statistics of experimental results.

Experimental result (image size=512 × 512)	No. of tampered pixels	No. of detected pixels (detection ratio)	False acceptance ratio (%)	False rejection ratio (%)
Exp. 3 shown in Fig. 11	5886	5379 (91.39%)	8.61	0
Exp. 4 shown in Fig. 12	57 037	48 491 (85.02%)	14.98	0

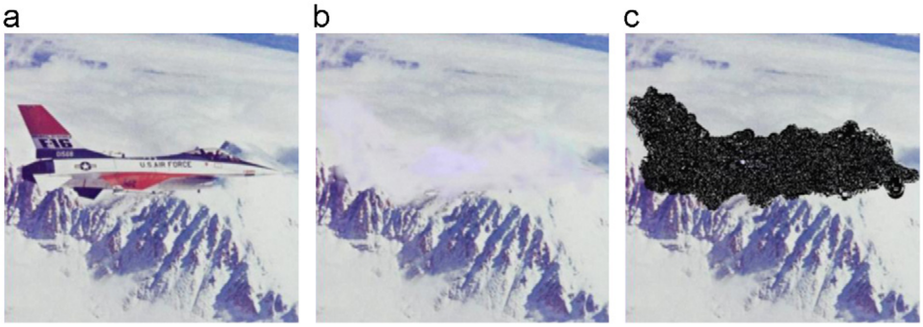


Fig. 13. Authentication result of an attacked stego-image in which the jet has been smeared: (a) Stego-image with authentication signals; (b) Modified stego-image; (c) Result with altered pixels detected and marked in black.

Table 8
Comparison of existing color image authentication methods.

	Losslessness of the image content	Tampering localization capability	Level of localization precision	detection of chrominance modification	Channel of containing embedded data	Tolerance of incidental image compression
Lu and Liao [1]	No	Yes	Block-level	No	Illumination channel	Yes
Peng & Liu [24]	No	Yes	Block-level	Yes	RGB channels	No
Byun et al. [25]	No	Only when alteration made in the blue channel	Pixel-level	Yes	Blue channel	No
Proposed method	Yes	Yes	Pixel-level	Yes	Alpha channel	Free from the misgiving

Table 9
Comparison of existing color image authentication method having the pixel-level authentication precision.

	False acceptance ratio (%)	False rejection ratio (%)
Method of [25]	≈ 50	0
Proposed method	≈ 12.5	0

and false acceptance and rejection ratios. Future studies may be directed to applications of the proposed method to copyright protection, digital rights management, recovery of altered image contents, etc.

References

[1] C.S. Lu, H.Y.M. Liao, Multipurpose watermarking for image authentication and protection, IEEE Transactions on Image Processing 10 (10) (2001) 1579–1592.

[2] C.W. Lee, W.H. Tsai, A secret-sharing-based method for authentication of grayscale document images via the use of the png image with a data repair capability, IEEE Transactions on Image Processing 21 (1) (2012) 207–218.

[3] G.J. Yu, C.S. Lu, H.Y.M. Liao, Mean quantization-based fragile watermarking for image authentication, Optical Engineering 40 (7) (2001) 1396–1408.

[4] C.W. Lee, W.H. Tsai, Optimal pixel-level self-repairing authentication method for grayscale images under a minimax criterion of distortion reduction, Optical Engineering 51 (5) (2012). 057006-1-057006-10.

[5] J.O. Ruanaidh, T. Pun, Rotation, scale and translation invariant spread spectrum digital image watermarking, Signal Processing 66 (3) (1998) 303–317.

[6] M. Barni, F. Bartolini, T. Furon, A general framework for robust watermarking security, Signal Processing 83 (10) (2003) 2069–2084.

[7] C. Deng, X. Gao, X. Li, D. Tao, A local Tchebichef moments-based robust image watermarking, Signal Processing 89 (8) (2009) 1531–1539.

[8] X. Gao, X. Li, D. Tao, C. Deng, J. Li, Robust reversible watermarking via clustering and enhanced pixel-wise masking, IEEE Transactions on Image Processing 21 (8) (2012) 3598–3611.

[9] I.S. Lee, W.H. Tsai, A new approach to covert communication via PDF Files, Signal Processing 90 (2) (2009) 557–565.

[10] T.Y. Liu, W.H. Tsai, A new steganographic method for data hiding in Microsoft Word documents by a change tracking technique, IEEE Transactions on Information Forensics and Security 2 (1) (2007) 24–30. 2007.

[11] A. Shamir, How to share a secret, Communication of the ACM 22 (11) (1979) 612–613.

[12] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding, IBM Systems Journal 35 (3–4) (1996) 313–336.

[13] J. Mielikainen, LSB matching revisited, IEEE Signal Processing Letters 13 (5) (2006) 285–287.

[14] C.H. Yang, C.Y. Weng, S. Wang, H.M. Sun, Adaptive data hiding in edge areas of images with spatial LSB domain systems, IEEE Transactions on Information Forensics and Security 3 (3) (2008) 488–497.

[15] Y.N. Wang, A. Pearmain, Blind image data hiding based on self reference, Pattern Recognition Letters 25 (15) (2004) 1681–1689.

[16] X. Gao, L. An, Y. Yuan, D. Tao, X. Li, Lossless data embedding using generalized statistical quantity histogram, IEEE Transactions on Circuits and Systems for Video Technology 21 (8) (2011) 1061–1070.

[17] S.H. Wang, Y.P. Lin, Wavelet tree quantization for copyright protection watermarking, IEEE Transactions on Image Processing 13 (2) (2004) 154–165.

[18] C.M. Pun, A novel DFT-based digital watermarking system for images, in: Proceedings of 8th International Conference on Signal Processing, Guilin, Yunnan, China, 2006, pp. 1245–1248.

[19] S. Pereira, T. Pun, Robust template matching for affine resistant image watermarks, IEEE Transactions on Image Processing 9 (6) (2000) 1123–1129.

[20] L.S.T. Chen, S.J. Lin, J.C. Lin, Reversible JPEG-based hiding method with high hiding-ratio, International Journal of Pattern Recognition. and Artificial Intelligence 24 (2010) 1–23.

[21] J.H. Lee, M.Y. Wu, Reversible Data-hiding method for palette-based images, Optical Engineering 47 (2008). 047008-1-047008-9.

[22] C.C. Lin, W.H. Tsai, Secret image sharing with steganography and authentication, Journal of Systems and Software 73 (2004) 405–414.

[23] R. Sakuldee, S. Udomhunsakul, Objective performance of compressed image quality assessments, International Journal of Computer Science 2 (4) (2007) 258–267.

[24] Z. Peng, W. Liu, Color image authentication based on spatiotemporal chaos and SVD, Chaos, Solitons and fractals 36 (2008) 946–952.

[25] S.C. Byun, I.L. Lee, T.H. Shin B.H. Ahn, A public-key based watermarking for color image authentication, in: IEEE International Conference on Multimedia and Expo, vol. 1, 2002, pp. 593–596.

[26] K.S. Kim, M.J. Lee, H.Y. Lee, H.K. Lee, Reversible data hiding exploiting spatial correlation between sub-sampled images, Pattern Recognition 42 (11) (2009) 3083–3096.

[27] Zhenyong Lixin Luo, Ming Chen, Xiao Chen, Zeng, Zhang Xiong, Reversible Image Watermarking Using Interpolation Technique, IEEE Transactions on Information Forensics and Security 5 (1) (2010) 16–21.

[28] Y. Hu, H.K. Lee, J. Li, DE-based reversible data hiding with improved overflow location map, IEEE Transactions on Circuits Systems and. Video Technology 19 (2) (2009) 250–260.

[29] C.C. Lin, N.L. Hsueh, A lossless data hiding scheme based on three-pixel block differences, Pattern Recognition 41 (4) (2008) 1415–1425.