A New Collage Steganographic Algorithm using Cartoon Design

Shuang Yi, Yicong Zhou*, Chi-Man Pun and C. L. Philip Chen

Department of Computer and Information Science,University of Macau, Macau, 999078, China

ABSTRACT

Existing collage steganographic methods suffer from low payload of embedding messages. To improve the payload while providing a high level of security protection to messages, this paper introduces a new collage steganographic algorithm using cartoon design. It embeds messages into the least significant bits (LSBs) of color cartoon objects, applies different permutations to each object, and adds objects to a cartoon cover image to obtain the stego image. Computer simulations and comparisons demonstrate that the proposed algorithm shows significantly higher capacity of embedding messages compared with existing collage steganographic methods.

Keywords: collage steganography, cartoon design, matrix encoding.

1. INTRODUCTION

Steganography came from a Roman word which means "writing in hiding".[1] It was originally used in the ancient war. A message was tattooed on a shaved slave's head, then it was hidden and delivered after his hair grew back.[2] Nowadays, with the development of the Internet, communication security has been an increasingly important subject, and steganography is widely used to protect secret or even classified information through open networks[3] and its concept can also be used for image encryption in some cases.[4]

Various steganographic methods were proposed in recent years. They mainly focus on embedding messages by modifying the statistical features of the cover media (i.e. an image) either in the spatial domain or in the transform domain using different technologies. Examples include the optimal pixel adjustment process (OPAP),[5] adaptive pixel pair matching (APPM),[6] least significant bit (LSB) substitution and pixel-value differences[7] in the spatial domain; and embedding messages in the transformed domain such as embedding messages in the sub-images decomposed by Fibonacci,[8] or by changing the JPEG quantization tables (QTs),[9] modifying the DCT or DWT coefficients[10,11] or altering header files[12] of the cover image. They intend to hide a number of messages into the cover media in such a way that minimizes the effect to the cover media. As a result, unauthorized users have difficulty to detect the existence of messages.[13,14] Other steganographic methods such as embedding messages in fractal parameters[15] and contrast/scaling, brightness/shifting fractal code coefficients[16] are also proposed in recent years.

Collage steganographic method, on the other hand, embeds messages into the cover media in an opposite way. It hides messages by modifying the appearance of cover media (i.e. an image) in a nature way, such as adding objects into the cover image.[17–19]

Shirali-Shahreza[18] proposed a collage steganography which embeds messages in three parameters of an object to be added to a cover image. These parameters include the object type, x-shifting and y-shifting positions. This method can embed only a few bits of messages into a cover image. Later Chen et al.[17] improved this method's payload of embedding messages by including two additional parameters: the rotation and scaling factors. However, its improved message payload is quite low. Furthermore, these two methods choose the cover image full of complex contents (e.g., a nature image). This results in small spaces for adding new objects. The object selection is extremely restricted by the cover image. For example, the content, size, and even light conditions of the selected objects should be fit to the nature of the cover image. Lee et al. proposed another object based steganographic method using the affine transform.[19] This method embeds messages in four coefficients of the affine transform. Its payload of embedding messages in one object is slightly larger than the former two methods but still limited. Meanwhile, its message extraction process is complicate and time-consuming. All these show that existing collage steganographic methods have a large room for further improvement.

In this paper, we propose a new collage steganographic algorithm combining the cartoon design with the LSB embedding and permutation technologies. The objective is to improve the payload and security of existing object-based

*Yicong Zhou: E-mail: yicongzhou@umac.mo, Telephone: +853 83978458

Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications 2014, edited by Reiner Creutzburg, David Akopian, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 9030, 90300N

steganographic methods. Experiments and comparisons will be provided to show the performance of our proposed algorithm.

The rest of this paper is organized as follows: Section 2 will briefly review several relevant techniques which will be used in the new collage steganographic algorithm introduced in Section 3. Section 4 will provide experimental results and compare our proposed algorithm with several existing collage steganographic methods. Some security analysis of the proposed algorithm will be discussed in Section 5. Section 6 draws a conclusion.

2. PRELIMINARY

In this section, two image databases and relevant techniques are briefly reviewed. They will be used for message embedding and extraction in our collage steganographic algorithm proposed in Section III.

2.1 Image Database Preparation

Before presenting our new method, two image databases need to be prepared, one is for the cover images (cartoon scene images) and the other is for the object images.

A cover image acts as a background on which objects will be pasted. We use $C = \{(C_i^r, C_i^g, C_i^b), i = 1, 2, \cdots, k\}$ to denote a set of k different colors in a cover image, where C_i^r, C_i^g and C_i^b represent the red, green and blue values of a color in the cover image, respectively. In order to easily extract objects from a steg image, we set $k < 16$.

The object image database contains S number of object images with the PNG or GIF format. All object images are transparent and contain an irregular shape so that they can be perfectly integrated with a cover image.

Suppose an object image O is of the size $M \times N$, we use $O_{i,j}^r$, $O_{i,j}^g$ and $O_{i,j}^b$ to denote the red, green and blue values of a pixel at location (i, j), respectively, where $i \in [1, M]$, $j \in [1, N]$, $O_{i,j}^r$, $O_{i,j}^g$, $O_{i,j}^b \in [0, 255]$. All pixels in an object image, except for pixels in the transparent areas, are called the foreground pixels, denoted by $O_{i,j}^*$:

$$O_{i,j}^* = \{(O_{i,j}^r, O_{i,j}^g, O_{i,j}^b) | O_{i,j}^r + O_{i,j}^g + O_{i,j}^b > 0\} \tag{1}$$

Otherwise, they are called the background pixels. The object images in the database are named by their index numbers from 1 to S.

2.2 Matrix Encoding

Matrix encoding was first implemented by Andreas Westfeld[20] and it is described as follows: Suppose we want to embed k message bits $x = (x_1, x_2, ..., x_k)$ in n ($n = 2^k - 1$) code word bits $a = (a_1, a_2, ..., a_n)$, x_1 and x_k denote the most significant bit (MSB) and LSB, respectively. The changed code word a' is calculated by[20]

$$a' = \begin{cases} a & \text{if } s = 0, \\ (a_1, a_2, ..., \overline{a_s}, ..., a_n) & otherwise \end{cases} \tag{2}$$

where

$$s = x \oplus f(a) \tag{3}$$

and

$$f(a) = \bigoplus_{1 \le i \le n} a_i \cdot i \tag{4}$$

'$\overline{a_s}$' and '$\oplus$' indicate logical negation of a_s and XOR operators, respectively.

2.3 Object Paste

In collage steganography, we embed messages by pasting objects in a cover image. To paste an object, we need to replace the object background pixels with the cover image's pixels where the object is to be pasted. Fig. 1 shows an example of pasting an object. Firstly, the object image (Fig. 1(a)) is separated into two sub-images: its foreground (Fig. 1(b)) and background (Fig. 1(c)). For a given location, a region with the same size of the object image is selected from the cover image. This region is multiplied with the object background subimage (Fig. 1(c)) and then add with the foreground subimage (Fig. 1(b)), obtaining a stego image with the pasted object as shown in Fig. 1(d).

(a) (b) (c) (d)

Figure 1. Paste an object in a cover image. (a) the object image; (b) the object foreground sub-image; (c) the object background sub-image; (d) a stego image with a pasted object.

3. PROPOSED METHOD

Motivated by cartoon design, which adds cartoon objects in a cartoon scene to form a visual meaningful cartoon image, this section proposes a new collage steganographic algorithm based on cartoon images.

3.1 Proposed algorithm

The goal of steganography is to conceal secrete messages in a media so that the third party has no idea about the existence of such messages. The fundamental underlying of our proposed algorithm is to embed the secrete messages into the LSB of the object foreground pixels, and add objects to a cartoon scene in order to generate the stego cartoon image. The appearance of the stego cartoon image changes with different scenes and objects.

The flow chart of our proposed algorithm is illustrated in Fig. 2. Here messages are converted to binary sequences. Each of them consists of a series of 0s and 1s. To ensure that the embedded messages correctly extracted by authorized receivers, a special byte, representing the message length, is added to the front of the messages. To begin with message embedding, a cartoon scene image is chosen from the cover image database, on which objects are pasted. Then an object image is selected from the object image database. Before message embedding, the sender may resize or rotate the object image to fit the nature of the cover image and other existing objects if necessary.

3.1.1 Message embedding

For a selected object image, only these foreground pixels are used to embed messages. The order of message embedding starts with the first foreground pixel in the object image, and moves from top to bottom and then left to right pixel by pixel as shown in Fig. 3. An illustration of an object image is shown in Fig. 3(a). The gray cells with labeled numbers represent the foreground pixels while the white cells in the transparent areas are the background pixels. We use (l_r, l_g, l_b) to denote the LSBs of the red, green, and blue components of a foreground pixel $O^*_{i,j}$ in an object image, where l_r, l_g, $l_b \in [0, 1]$. A special case of matrix encoding (i.e., k = 2 and n = 3 in Section 2.2) is applied to embed messages. For three LSBs $a = (l_r, l_g, l_b)$ of a color pixel, two message bits $x = (m_2, m_1)$ are embedded based on Eqns. (2)-(4). It can be simplified to the embedding rules as follows:

- $m_1 = l_r \oplus l_b, m_2 = l_g \oplus l_b \Rightarrow (l_r, l_g, l_b)$
- $m_1 = \overline{l_r} \oplus l_b, m_2 = l_g \oplus l_b \Rightarrow (\overline{l_r}, l_g, l_b)$
- $m_1 = l_r \oplus l_b, m_2 = \overline{l_g} \oplus l_b \Rightarrow (l_r, \overline{l_g}, l_b)$
- $m_1 = l_r \oplus \overline{l_b}, m_2 = l_g \oplus \overline{l_b} \Rightarrow (l_r, l_g, \overline{l_b})$

where $\oplus$ denotes the XOR operator.

For example, suppose we want to embed two message bits 'm_2m_1' = '11' into a pixel with its color component values (191, 128, 93), we first get its LSBs (1, 0, 1). According to the embedding rules, $1 = \overline{1} \oplus 1$, $1 = 0 \oplus 1$, the algorithm

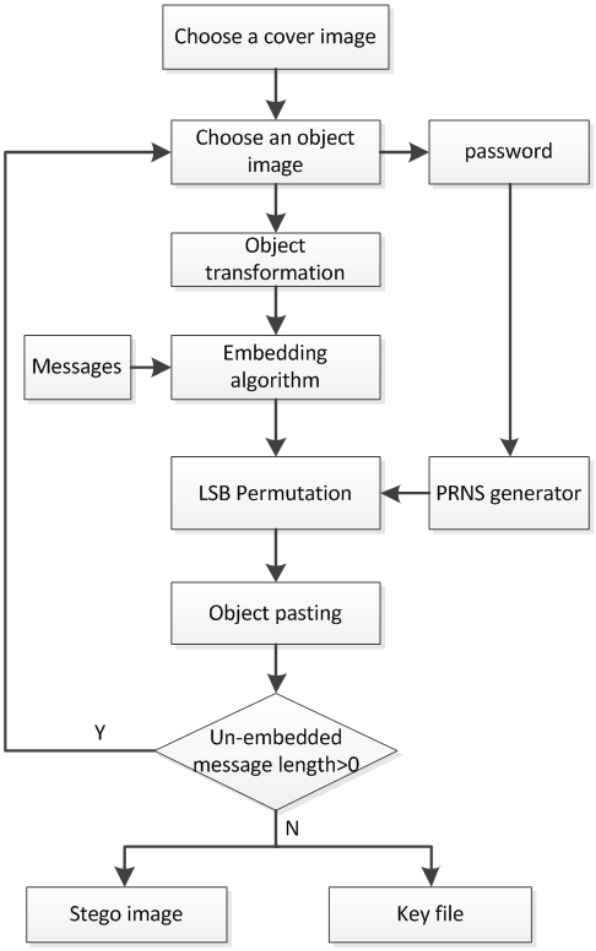


Figure 2. Flow chart of the proposed collage steganographic algorithm

changes the first LSB bit 1 into 0 while keeping the second and third bits unchanged. The pixel value (191, 128, 93) changes to (190, 128, 93) with two message bits embedded.

If a pixel value falls into the data range of the background pixels after embedding message bits, this pixel, such as the pixels with a red dash square in Fig. 3(b), will be discarded. The message bits will be re-embedded in the next foreground pixel.

(a) (b)

Figure 3. Illustration of embedding messages in an object image. (a) 35 foreground pixels; (b) 30 foreground pixels with several pixels discarded during messages embedding.

After message embedding, a pixel permutation is applied to these LSBs of the foreground pixels in the object image (e.g., gray cells with labeled numbers in Fig. 3(b)). The permutation sequence is derived from a pseudo-random number

sequence (PRNS) generator using the index number of the object image in its database as the security key. Thus, different objects uses different PRNSs for pixel permutation, achieving a high level of security.

An object embedded with messages will be pasted on a cover image (a cartoon scene) with the location given by the sender. Selection of objects, cover images (cartoon scenes), and object locations is indeed a science of art and follows the basic law of nature because we can neither paste a dog in the sky nor put the sun on the grass. Therefore, even for the same number of objects pasted on the same cartoon scene, different object locations may result in completely different stego images.

Repeating the above embedding process until all messages are embedded, we finally obtain the stego image and a key file containing the object locations and the number of colors in the selected cover image (cartoon scene). Both the stego image and key file need to be sent to the receiver for extracting messages. Note that a new object to be pasted is allowed neither to overlap any existing object in the cover image nor to be out of the boundary of the cover image. Otherwise the object location must be reset.

3.1.2 Message extraction

This subsection shows how to extract messages from a stego image based on the key file. The flow chart of the message extraction procedures is shown in Fig. 4.

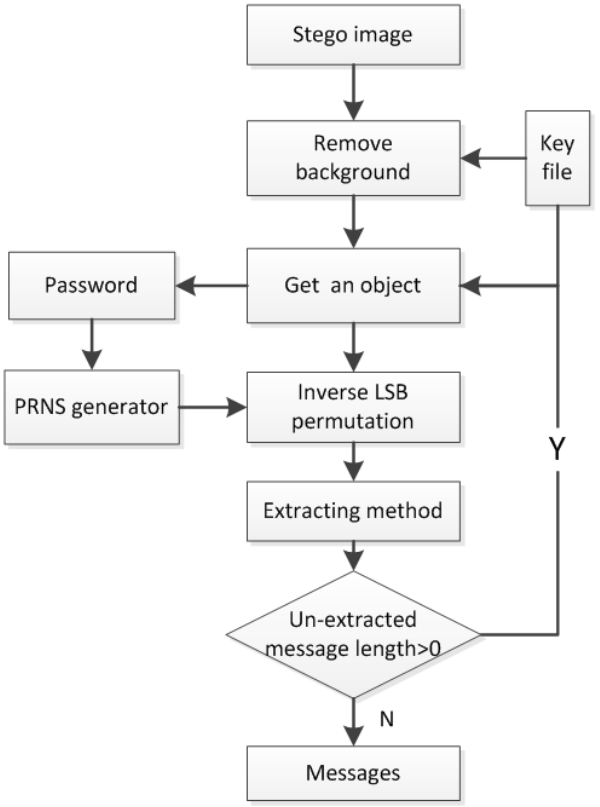


Figure 4. Flow chart of message extraction

After removing the background colors from a stego image, we obtain objects sequentially based the key file. This ensures that messages will be combined in a correct order. To extract messages from an extracted object, the object's index number needs to be extracted and used as a password P^* to generate inverse PRNS. To do this, a set of seven invariant moments[21] is used. Because objects may be modified before message embedding, these moments invariants under transformation, scaling and rotation, can be used to recover the password P^* which is obtained by

$$P^* = \arg\min_P f(O_P), \quad 1 \le P \le S \tag{5}$$

where

$$f(O_P) = \frac{1}{7} \sum_{1 \le i \le 7} (\phi_i^P - \phi_i^O)^2 \tag{6}$$

O_P denotes the P^{th} object in the object image database which contains S number of object images; ϕ_i^P and ϕ_i^O indicate the i^{th} invariant moment values of the P^{th} object in the object image database and the O^{th} object in the stego image, respectively.

Using the sequence obtained by the PRNS generator with the password P^*, an inverse LSB permutation process is applied to the object's foreground LSBs to recover the original LSBs. The message bits can be extracted using the rules defined below:

- $m_1 = l_r \oplus l_b$
- $m_2 = l_g \oplus l_b$

For example, for a pixel (190, 128, 93) in a stego object, the LSBs (l_r, l_g, b_b) = (0, 0, 1). We get $m_1 = 0 \oplus 1 = 1$ and $m_2 = 0 \oplus 1 = 1$, two original message bits '11' are successfully extracted.

After all message bits are extracted, messages are combined sequentially based on the object paste order obtained from the key file.

3.2 Discussion

In summary, our proposed algorithm has at least following advantages:

- Compared with existing collage steganographic methods,[17,18] which embed messages in the object's parameters (i.e. the object type, x-shifting, y-shifting, rotation and scaling), our proposed algorithm embeds messages in the LSBs of objects and significantly improves their payload of embedding messages.
- The cover images of existing collage steganographic methods are nature images full of complex contents. This significantly restricts the flexibility of selecting the appropriate regions to paste objects. Our proposed algorithm, on the other hand, uses the cartoon scenes as the cover images. Thus, any location in the cover image can be used to paste objects. Hence it increases the number of objects to be pasted.
- The cover and object images in our algorithm are cartoons. This offers users great design flexibility and simplicity. The designed stego image has natural view without considering nature conditions such as shadows and light sources
- Senders construct various scenarios by manually choosing objects and the stego images are visually meaningful.
- We can consider the cover images as special objects and embed messages on their colors and thus the payload of embedding messages will be further increased.

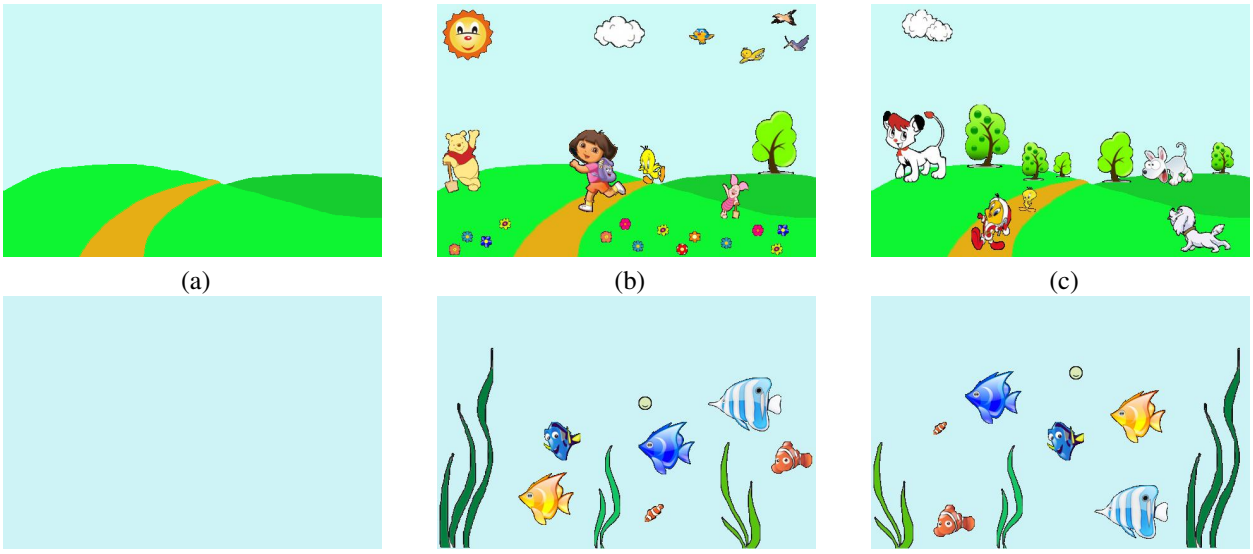
4. EXPERIMENTAL RESULTS AND COMPARISONS

This section demonstrates the performance of the proposed algorithm and compares it with existing collage steganographic methods.

Fig. 5 shows four examples of stego images generated by our proposed algorithm. These stego images have 7.6 KB messages embedded in two different cover images with size of $600 \times 400 \times 8$. In our experiments, the message is a bit sequence with a length of 28 bits.

As can be seen from the results in Fig. 5, for a given length of messages, if the sizes of objects are small, we need a large number of objects to embed the messages. For example, we use 11 objects in Fig. 5(e) and 25 objects in Fig. 5(c). Furthermore, we can construct different stego images by adding different objects (e.g., Figs. 5(b)-(c)) or the same objects with different locations (e.g., Figs. 5(e) and (f)). The generated stego images are nature and visually meaningful.

In addition, due to different object paste orders, the same stego images may contain different messages. Even the same stego images with the same messages, different object paste orders also result in different key files. This is extremely



(a) (b) (c)

(d) (e) (f)

Figure 5. Stego images with 7.6KB messages embedded by our proposed algorithm. (a) and (d) are two cover images; (b) and (c) are two stego images generated from (a) with different objects added; (e) and (f) are two stego images generated from (d) with 10 objects added at different locations.

important for message extraction. Without this correct key file, the receivers cannot extract the messages as well. It increases the security level of our algorithm consequently.

Table 1 compares the payload of embedding messages of our proposed algorithm with those of three existing collage steganographic methods. Our experimental settings for this comparison are: the object image database contains 25 object images, the permissible area is 30×20, the rotation factor is 3, the scaling factor is 2 and the average object size of our algorithm is $70 \times 70 \times 8$. From the results in Table 1, we know that, for a given size of the object or for the same number of objects, our proposed algorithm significantly improves the payload compared with other methods. Furthermore, for a given size of the cover image, the payloads of three existing methods heavily depend on the nature scenarios of the cover images. The complex nature of the cover image results in less spaces for adding new objects. Their payloads of each object are constant. However, in our algorithm, any location in the cover image can be used to add objects and an object with a bigger size has a larger payload of embedding messages. In addition, different permutations for different objects increase the security level of messages as well.

Table 1. Payload comparison of different collage steganographic methods

Methods	**Payload in an $30 \times 30 \times 8$ object (bits)**	**Payload in 8 objects in objects in a cover image (bits)**
Shirali's algorithm[18]	13	< 130
Chen's algorithm[17]	16	130
Lee's algorithm[19]	44	< 400
Proposed algorithm	890	39000

5. SECURITY ANALYSIS

As can be seen, even though the embedding algorithm is known to the public, it is difficult for hackers to extract the embedded data from stego images since we adopt two mechanisms to enhance the security. Firstly, message embedding is based on the order of object selection. This means that the same object with different selection orders contains totally different embedding data. Our algorithm scrambles the messages using the object selection order which is recorded in the key file. Therefore, it ensure the security of messages. Secondly, a LSB permutation operation is applied to change the

message locations within objects. The password used to generate permutation sequence is obtained from the object image database. It is shared by the senders and receivers. Without this object image database, hackers are extremely difficult to extract the messages as well.

Here we show the LSB and histogram analysis to both original objects and stego objects, PSNR is also used to analysis stego objects with different message embedding rate.

5.1 LSB plane analysis

Fig. 6 shows the LSB in R, G and B planes of original and stego objects. As can be seen, the stego object (Fig. 6(e)) has no difference with the original object (Fig. 6(a)) in human color vision, and the object regions' LSB pixels of R, G, and B color planes (Fig. 6(f)-(h)) are randomly distributed within the object regions after message embedding.

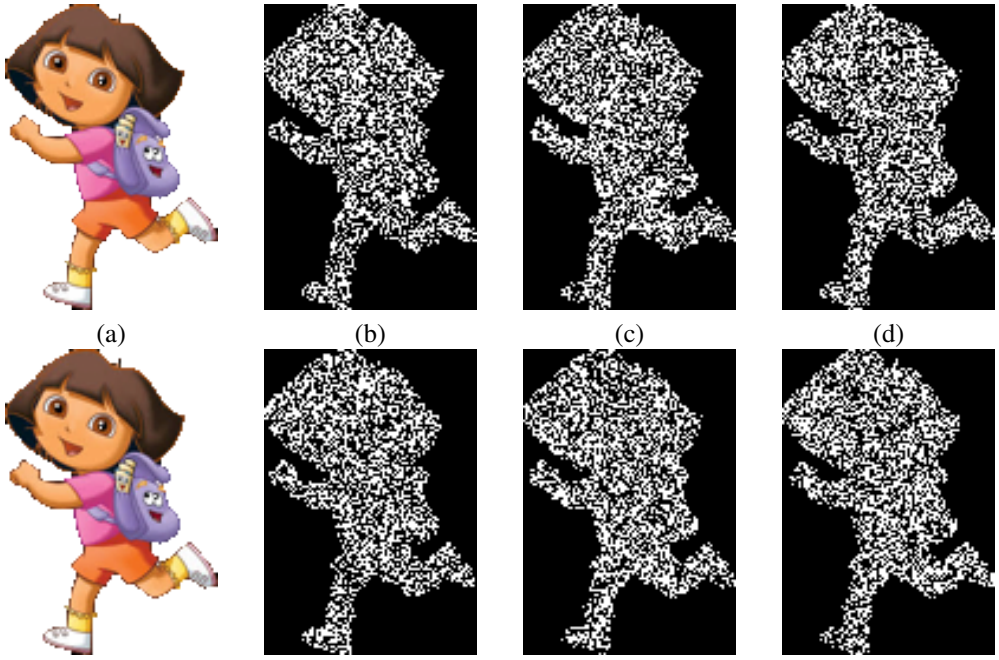


Figure 6. LSB comparison of an object with and without messages embedded. (a) the original clean object; (b)-(d) show the LSBs of the image (a) in R, G, B color planes, respectively; (e) the stego object with messages embedded; (f)-(g) show the LSBs of image (e) in R, G, B color planes, respectively.

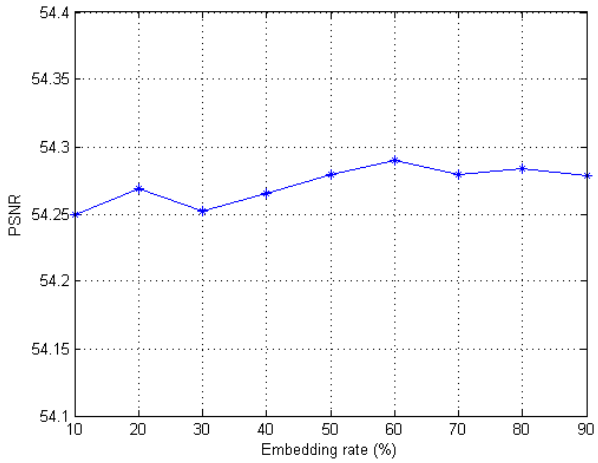


Figure 7. PSNR of stego object images with different message embedding rate

5.2 PSNR analysis

Fig. 7 plots the PSNR results of stego images with different message embedding rates. The results indicate that the stego images have high PSNR values no matter how much the message embedding rate is because of the LSB modification. On the other hand, the PSNR values are within a small data range without rapidly dropping down or rising up due to the LSB permutation after message embedding. Thus, the hackers have extremely difficulty to disclose the originally embedded message even the size of message.

5.3 Histogram analysis

Fig. 8 shows the object image's histogram of R, G and B color planes both with and without messages embedded. From the result we know that, colors slightly changed after messages embedded in the stego object image (Fig. 8 (j)-(l)). Fig. 8 (i) shows that, the R, G, and B color planes' LSB modify positions are randomly distributed in the stego object image because of the LSB permutation, and hackers cannot get the correct messages without the password which is used to generate the permutation sequence.

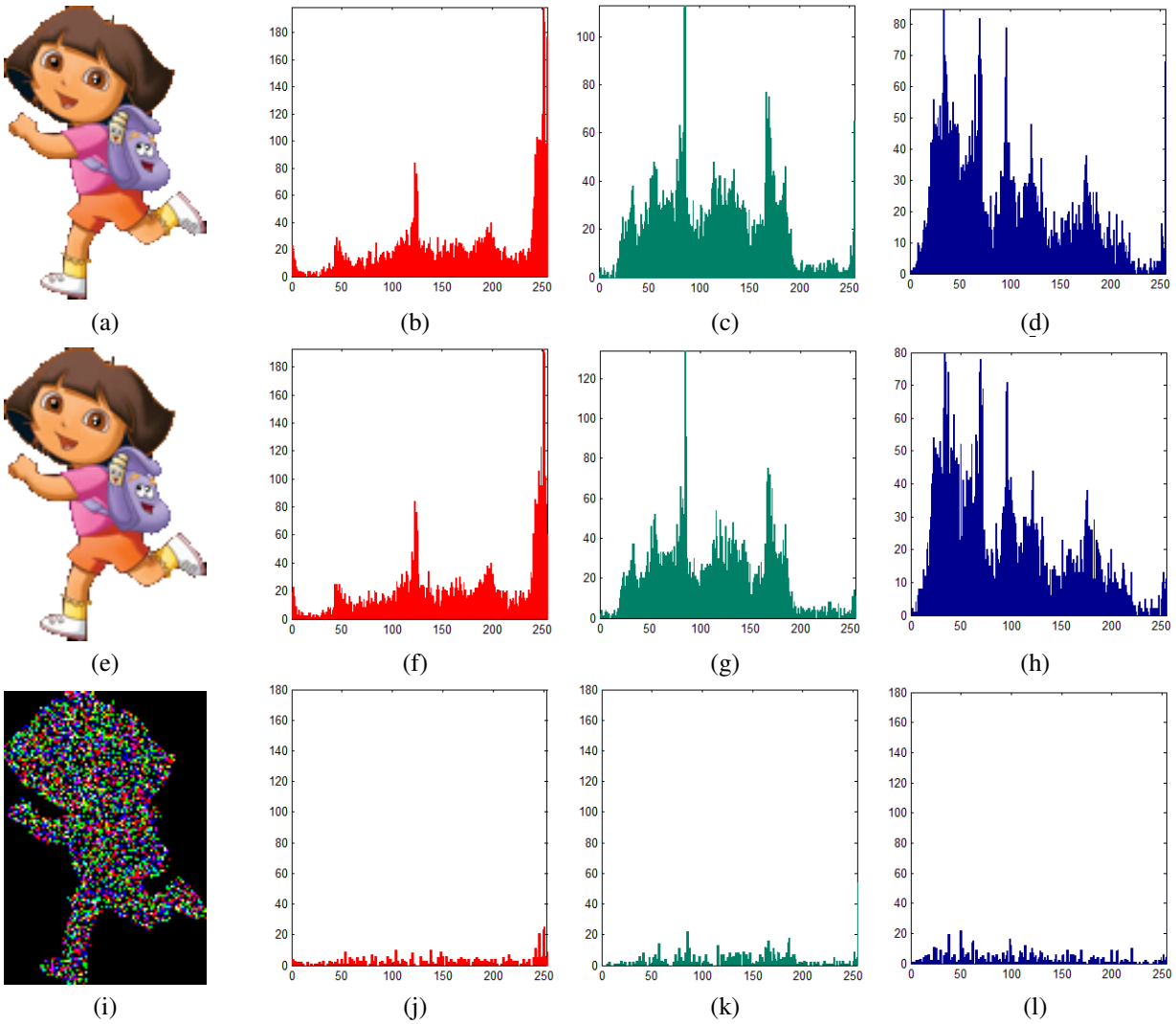


Figure 8. Histgram comparison of an object with and without messages embedded. (a) the original object; (b)-(d) show histograms of the R, G, B color planes of the image (a), respectively; (e) the stego object with messages embedded; (f)-(g) show histograms of the R, G, B color planes of the image (e), respectively; (i) the LSB differences between (a) and (e); (j)-(l) show the histogram differences of R, G and B color planes between (a) and (e), respectively.

6. CONCLUSION

In this paper, a new collage steganographic algorithm was introduced. It embeds messages in cartoon objects and then these objects are pasted in a cartoon cover image to obtain the final stego image. The proposed algorithm uses different permutations to shuffle the massages within each object, protecting messages with a high level of security. Experiment results have shown that our proposed algorithm significantly improves the payload of embedding messages and while maintaining excellent appearances of stego images than other collage steganographic methods.

ACKNOWLEDGMENTS

This work was supported in part by the Macau Science and Technology Development Fund under Grant 017/2012/A1 and by the Research Committee at University of Macau under Grants SRG007-FST12-ZYC, MYRG113(Y1-L3)-FST12-ZYC and MRG001/ZYC/2013/FST.

REFERENCES

[1] Shirali-Shahreza, S. and Shirali-Shahreza, M., "Improved collage steganography," in [*the 4th International Conference on Emerging Technologies*], 223–227, 2008.

[2] Provos, N. and Honeyman, P., "Hide and seek: an introduction to steganography," *IEEE Security and Privacy* **1**(3), 32–44 (2003).

[3] Agaian, S. S., "Steganography & steganalysis, an overview of research & challenges," *NATO Science for Peace and Security Series - D: Information and Communication Security* **17**, 179–210 (2008).

[4] Zhou, Y. and Agaian, S., "Image encryption using the image steganography concept and PLIP model," in [*System Science and Engineering (ICSSE), 2011 International Conference on*], 699–703 (2011).

[5] Yang, C. H., "Inverted pattern approach to improve image quality of information hiding by LSB substitution," *Pattern Recognit* **41**(8), 26742683 (2008).

[6] Hong, W. and Chen, T. S., "A novel data embedding method using adaptive pixel pair matching," *IEEE Transactions on Information Forensics and Security* **7**(1), 176–184 (2012).

[7] Khodaei, M. and Faez, K., "New adaptive steganographic method using least significant-bit substitution and pixel-value differencing," *IET, Image Processing* **6**(6), 677–686 (2012).

[8] Agaian, S. S., Cherukuri, R. C., and Sifuentes, R., "A new secure adaptive steganographic algorithm using Fibonacci numbers," in [*Region 5 Conference, 2006 IEEE*], 125–129 (2006).

[9] Guo, J. M. and Thanh-Nam, L., "Secret communication using JPEG double compression," *IEEE Signal Processing Letters* **17**(10), 879–882 (2010).

[10] Prabakaran, G., Bhavani, R., and Kanimozhi, K., "Dual transform based steganography using wavelet families and statistical methods," in [*International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME)*], 287–293, 2013.

[11] Huang, F. J., Huang, J. W., and Shi, Y. Q., "New channel selection rule for JPEG steganography," *IEEE Transactions on Information Forensics and Security* **7**(4), 1181–1191 (2012).

[12] Yildiz, Y. O., Panetta, K., and Agaian, S. S., "New quantization matrices for JPEG steganography," *Proceedings of SPIE* **6579**, 65790D–1–65790D–11 (2007).

[13] Hou, C. L., Lu, C. C., Tsai, S. C., and Tzeng, W. G., "An optimal data hiding scheme with tree-based parity check," *IEEE Transactions on Image Processing* **20**(3), 880–886 (2011).

[14] Filler, T., Judas, J., and Fridrich, J., "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security* **6**(3), 920–935 (2011).

[15] Agaian, S. S. and Susmilch, J. M., "Fractal steganography using artificially generated images," in [*Region 5 Conference, 2006 IEEE*], 312–317 (2006).

[16] Chen, M.-C., Agaian, S. S., Chen, C. L. P., and Rodriguez, B. M., "Image steganography in fractal compression," *Proc. SPIE* **7351**, 73510Y–73510Y–12 (2009).

[17] Chen, M. C., Agaian, S. S., and Chen, C., "Generalized collage steganography on images," in [*IEEE International Conference on Systems, Man and Cybernetics*], 1043–1047, 2008.

[18] Shirali-Shahreza, M. and Shirali-Shahreza, S., "Collage steganography," in [*5th IEEE/ACIS International Conference on Computer and Information Science.*], 316–321, 2006.

[19] Lee, Y. K. and Chen, L. H., "Object-based image steganography using affine transformation," in [*International Journal of Pattern Recognition and Artifcial Intelligence*], **16**(6), 681–696, 2002.

[20] Westfeld, A., "F5 - a steganographic algorithm: High capacity despite better steganalysis," in [*4th International Workshop on Information Hiding*], 289–302, 2001, Springer-Verlag.

[21] Gonzalez, R. C. and Woods, R. E., "Digital image processing," in [*Prentice Hall, NJ, 3rd ed*], 861–864, 2008.