A New Steganographic Method Based on Information Sharing via PNG Images[†]

Che-Wei Lee
Institute of Multimedia Engineering
National Chiao Tung University, Hsinchu, Taiwan
lcw.cs94g@nctu.edu.tw

[*]Wen-Hsiang Tsai
Department of Computer Science
National Chiao Tung University, Hsinchu, Taiwan
whtsai@cis.nctu.edu.tw

***Abstract*—A new steganographic method via PNG images based on the information sharing technique is proposed. The coefficients of the polynomial functions of the Shamir's (*k*, *n*)-threshold secret sharing method are utilized as carriers of a given secret data string to be hidden to generate shares. The shares then are embedded into the alpha-channel plane of a cover PNG image. The resulting stego-image with perceptible white noise is further removed by mapping computed share values into suitable ranges. Appropriate measures for enhancing the security of the proposed method are also described. Experimental results show the effect of the proposed method.**

Keywords – steganography, data hiding, information sharing, PNG image.

I. INTRODUCTION

Steganography is a kind of data hiding technique and can be applied to applications like covert communication, secret keeping, etc. People hide a secret message into a *cover file*, resulting in a *stego-file*; and a receiver can extract the hidden message from the stego-file to complete the communication or the retrieval of the secret message.

On the other hand, the *information sharing* method is developed to protect the security of concerned message data. In the method, a secret message is processed to construct several *shares* which are then distributed to a number of participants to keep. Conventionally, data hiding and information sharing are two irrelevant issues in the domain of information security research. In this study, a new data hiding method with steganographic effects based on the concept of information sharing is proposed for hiding secret data into PNG (portable network graphics) images.

Many data hiding methods based on the spatial domain [1-3] and frequency domain [4-6] have been proposed. From another viewpoint, many types of images have been used as cover media, like BMP, JPEG, and GIF images [7-9]. Lee et al. [7] proposed a data hiding method based on palette modification for BMP images. Wong et al. [8] embedded data into JPEG images using the randomized parity feature in selected DCT coefficients of the cover image. Tzeng, Yang, and Tsai [9] hid secret data into GIF image pixels by color ordering and mapping.

In the proposed information-sharing-based data hiding method, a PNG image is given 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 theO beginning of the proposed data hiding process. A data string to be hidden is transformed next into shares by the Shamir's secret sharing method, which are 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 message 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 selected skillfully. Also, a mapping function is designed for adjusting the alpha-channel values to create *uniform transparency* in the alpha-channel plane, resulting in a higher steganographic effect in the stego-image. The R, G, and B channels are untouched so that the original image appearance revealed by the color information of these three channels is kept. It is noted that this method is not an information sharing technique, but a method of steganography via data hiding.

The remainder of this paper is organized as follows. In Section II, the Shamir method on which the proposed data hiding method is based is reviewed first. In Section III, the details of the proposed method, including the data embedding and extraction processes, are described. Experimental results are shown in Section IV. Finally, some conclusions are made in Section V.

II. REVIEW OF 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 [10], where a secret *d* in the form of an integer is to be shared, *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 *d*. 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.

Steps:

1. Choose a prime number *p* randomly.

978-1-4244-5585-0/10/$26.00 ©2010 IEEE

[†]This work was supported financially by the Ministry of Economic Affairs under Projects Nos. MOEA 97-EC-17-A-02-S1-032 and MOEA 98-EC-17-A-02-S2-0047 in te Technology Development Programs for Academia.

[*]Also with Department of Information Communication, Asia University, Taiwan.

2. Select $k - 1$ integer values $c_1, c_2, \ldots, c_{k-1}$ within the range of 0 through $p - 1$.
3. Select n distinct real values $x_1, x_2, \ldots, x_n$.
4. Use the following $(k - 1)$-degree polynomial to generate n equations to compute n function values $F(x_i)$:

$$F(x_i) = (d + c_1x_i + c_2x_i^2 + \ldots + c_{k-1}x_i^{k-1})_{\bmod p}, \qquad (1)$$

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

Since there are k coefficients, including d and c_1 through c_{k-1}, in Eqs. (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 and the name, (k, n)-*threshold*, for the Shamir method [10]. Below is a description of such a way of equation solving for secret recovery in the form of an algorithm.

Algorithm 2: secret recovery.

Input: m shares in the form of $(x_j, F(x_j))$ collected from the n participants where $1 \le j \le n$, $k \le m \le n$, and k is the threshold mentioned in Algorithm 1.

Output: the secret d hidden in the shares.

Steps:

1. Collect any k of the m shares, say, $(x_{i_1}, F(x_{i_1}))$, $(x_{i_2}, F(x_{i_2}))$, $\ldots$, $(x_{i_k}, F(x_{i_k}))$ and use them to set up the following equations:

$$F(x_{i_j}) = (d + c_1x_{i_j} + c_2x_{i_j}^2 + \ldots + c_{k-1}x_{i_j}^{k-1})_{\bmod p}, \qquad (2)$$

where $j = 1, 2, ..., k$ and $1 \le i_j \le n$.
2. Solve the k equations above by Lagrange's interpolation to obtain the desired secret value d [11] as follows:

$$d = (-1)^{k-1}[F(x_{i_1})\frac{x_{i_2}x_{i_3}...x_{i_k}}{(x_{i_1}-x_{i_2})(x_{i_1}-x_{i_3})...(x_{i_1}-x_{i_k})}$$
$$+F(x_{i_2})\frac{x_{i_1}x_{i_3}...x_{i_k}}{(x_{i_2}-x_{i_1})(x_{i_2}-x_{i_3})...(x_{i_2}-x_{i_k})}$$
$$+...+F(x_{i_k})\frac{x_{i_1}x_{i_2}...x_{i_{k-1}}}{(x_{i_k}-x_{i_1})(x_{i_k}-x_{i_2})...(x_{i_k}-x_{i_{k-1}})}]_{\bmod p}.$$

3. Compute the values c_1 through c_{k-1} by expanding the following equation and compare the result with (2) in Step 1:

$$F(x) = [F(x_{i_1})\frac{(x-x_{i_2})(x-x_{i_3})...(x-x_{i_k})}{(x_{i_1}-x_{i_2})(x_{i_1}-x_{i_3})...(x_{i_1}-x_{i_k})}$$
$$+F(x_{i_2})\frac{(x-x_{i_1})(x-x_{i_3})...(x-x_{i_k})}{(x_{i_2}-x_{i_1})(x_{i_2}-x_{i_3})...(x_{i_2}-x_{i_k})}$$
$$+...+F(x_{i_k})\frac{(x-x_{i_1})(x-x_{i_2})...(x-x_{i_{k-1}})}{(x_{i_k}-x_{i_1})(x_{i_k}-x_{i_2})...(x_{i_k}-x_{i_{k-1}})}]_{\bmod p}.$$

Step 3 in the above algorithm is included additionally for the reason that we want to compute the values of c_i in the proposed method. In other applications, if only the secret need be recovered, this step may be eliminated.

III. Proposed Method for Data Hiding via PNG Images

Based on the Shamir method [10] described previously, the basic idea of the proposed method for hiding a given data string S in a cover PNG image I to yield a stego-image I' is described as follows.

(1) Transform S into a sequence of decimal numbers.
(2) Take sequentially a number of the resulting decimal numbers as the values of d and c_i in Eqs. (1) to compute *partial shares* $F(x_i)$.
(3) Embed $F(x_i)$ into I by replacing the alpha-channel values of I with those of $F(x_i)$.
(4) Repeat (2) until no more decimal number is left, resulting in a stego-image I'.

3.1 Proposed Algorithm for Data Embedding

The details of the ways we embed and extract the shares are described 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:

1. (*Initialization*) Divide M into t-bit segments with $t = 3$ and transform each segment into a decimal number, resulting in a decimal-number sequence $M' = d_1d_2d_3...$ where $0 \le d_i \le 7$.
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'.
3. (*Partial share creation*) Set p, c_i, and x_i in Eqs. (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 $x_4 = 4$,

   resulting in the following equations:

$$q_1 = F(x_1) = (m_1 + m_2x_1 + m_3x_1^2 + m_4x_1^3)_{\bmod p},$$
$$q_2 = F(x_2) = (m_1 + m_2x_2 + m_3x_2^2 + m_4x_2^3)_{\bmod p},$$
$$q_3 = F(x_3) = (m_1 + m_2x_3 + m_3x_3^2 + m_4x_3^3)_{\bmod p},$$
$$q_4 = F(x_4) = (m_1 + m_2x_4 + m_3x_4^2 + m_4x_4^3)_{\bmod p}. \qquad (4)$$

   and compute the partial shares of q_1 through q_4 accordingly.
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.
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 q_1' through q_4', respectively.
   5.2 Remove m_i through m_{i+3} from M'.

6. (*End of looping*) If M' is not empty, then go to Step 2 to process the next four decimal numbers 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 Eqs. (4) in Step 3 of the above algorithm and inserted in the alpha channels of I are between 0 and 10 because the prime value p used in Eqs. (4) is 11. And after performing Step 4 of the algorithm, the values of q_1' through q_4' form a small range of integers 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 large 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, which is undesired in steganographic applications like covert communication. 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 Eqs. (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 $t = 2$.

In addition, it is noted that Algorithm 3 takes every *four* decimal 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 decimal numbers each time and embeds them into n pixels. For this, just modify part of Step 3 to be

…

(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:

$$q_1 = F(x_1) = (m_1 + m_2x_1 + m_3x_1^2 + \ldots + m_nx_1^n)_{\text{mod } p},$$
$$q_2 = F(x_2) = (m_1 + m_2x_2 + m_3x_2^2 + \ldots + m_nx_2^n)_{\text{mod } p},$$
$$q_3 = F(x_3) = (m_1 + m_2x_3 + m_3x_3^2 + \ldots + m_nx_3^n)_{\text{mod } p},$$
$$\ldots$$
$$q_n = F(x_n) = (m_1 + m_2x_n + m_3x_n^2 + \ldots + m_nx_n^n)_{\text{mod } p}. \qquad (4')$$

This 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.

3.2 Proposed Algorithm for Data Extraction

Now, the process of hidden secret extraction is described in the following.

Algorithm 4: secret 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.

1. (*Initialization*) Create an empty string M.
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'.
3. Subtract 245 from each of q_1' through q_4' to obtain q_1 through q_4, respectively. Perform the secret recovery process described by Algorithm 2 to extract the values m_1 through m_4 of the decimal format hidden in q_1 through q_4.
4. Transform the extracted values of m_1 through m_4 into binary bits and append in order each of them to the end of M.
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.

IV. Security Considerations

To enhance the security of the embedded data in the stego-image, the following measures may be adopted.

(1) *Use of a random key* --- A key may be used to randomize the pixel positions for embedding the partial shares q_1 through q_k in Step 5 of Algorithm 3. In this way, the probability of correctly guessing the pixel positions is $1/[(n\times n)!]$ where $n\times n$ is the size of the cover image, which is very large for common image sizes like $n\times n = 512\times512 = 64K$.

(2) *Randomization of constants used in the proposed algorithms* --- The values of x_1 through x_n used in Step 3 of the generalized version of Algorithm 3 are designed to be constants. However, they actually may be chosen to be random, as long as their magnitudes are within the allowed range of integer values ($0 \leq x_i \leq 2^{16}$ for single-precision integers). The probability of correctly guessing all the values of x_1 through x_n is approximately $1/(2^{16})^n$, which is again very large even for a small $n = 4$.

(3) *Encryption of data to be embedded* --- The data to be embedded may be randomized using a key and a known encryption technique such as Sha-1, DES, etc.

(4) *Combinations of the above techniques* --- The above three techniques may be combined and used as a single security enhancement measure which is sufficient for most applications.

V. Experimental Results

A lot of experiments have been conducted to test the proposed algorithms. Some results using two test images, named Lena and Jet, as shown in Figs. 1(a) and 1(c), respectively, are presented here. The results of applying the proposed method using Algorithm 3 to embed a long sequence of binary message data into the two images are shown in Figs. 1(b) and 1(d), respectively. As can be seen from the figures, the steganographic effect is obvious — the stego-images are visually almost identical to the cover images, respectively, although the alpha-channel contents of the stego-images include embedded the message data. By the way, it is noted that in Algorithm 3, the value of t, which is the number of bits taken as a segment and transformed into a decimal number for use in the secret sharing process performed by Algorithm 3, was taken to be 3.

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 images Lena and Jet as cover images. Specifically, in Table 1 we show, in addition to the data hiding capacities, the qualities of the alpha-channel planes of the two corresponding stego-images in terms of the PSNR measure; and in Figure 2, we show the stego-images of Lena corresponding to $t = 1, 2, \ldots, 7$. The PSNR values of the alpha-channel planes of the stego-images were computed with the original alpha-channel values in the cover images being taken to be all 255. As can be seen, when $t = 3$, although the PSNR is at the level of 32.69 which is not high enough, yet the corresponding stego-image quality shown by Fig. 2(c) is still visually good. In addition, the corresponding data hiding capacity is $R = tS = 3\times512\times512 = 75K = 786432$ bits, which is good enough for general applications.

In fact, it can be seen from Fig. 2(d) that even when t is taken to be 4, the stego-image quality is still acceptable with the data hiding capacity increased to 100K 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. Note also that different from other methods, the channels of R, G, and B of the cover image is not processed by the proposed method, yielding a *lossless* result in the color channels.

VI. 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 by using the coefficients of some polynomial functions as data carriers for computing the shares. The alpha-channel plane of a cover PNG image is utilized to embed the partial shares, yielding a stego-image with undesirable white noise. The white noise is then eliminated by choosing a small prime number, dividing the input data string into 3-bit segments, and mapping computed share values into a range of alpha-channel values near their maximum value of 255. Generalization of the method to allow compromise between the resulting data hiding capacity and stego-image quality has also been carried out. And four measures for enhancing the security of the proposed method have been mentioned, namely, the use of a random key, randomization of constants used in the proposed algorithms, encryption of data to be embedded, and a combination of the three former measures. Good experimental results proving the effectiveness of the proposed methods in the aspects of data hiding capacity, steganographic effect, and stego-image quality have also been presented.

References

[1] R. Z. Wang, C. F. Lin and J. C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm," *Pattern Recognition*, vol. 34, no. 3, pp. 671-683, 2000.

[2] C. K. Chan and L. M. Chen, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, no. 3, pp. 469-474, 2004.

[3] H. Izadinia, F. Sadeghi and M. Rahmati, "A new steganographic method using quantization index modulation," *International Conference on Computer and Automation Engineering* , pp. 181-185, Mar. 2009.

[4] Y. N. Wang and A. Pearmain, "Blind image data hiding based on self reference," *Pattern Recognition Letters*, vol. 25, pp. 1681-1689, 2004.

[5] S. H. Wang and Y. P. Lin, "Wavelet tree quantization for copyright protection watermarking," *IEEE Transactions on Image Processing*, vol. 13, pp. 154-165, 2004.

[6] C. M. Pun, "A novel DFT-based digital watermarking system for images," *Proceedings of 8th International Conference on Signal Processing*, Guilin, Yunnan, China, vol. 2, pp. 1245-1248, Nov. 2006..

[7] J. H. Lee and M. Y. Wu, "An Iterative Method for Lossless Data Embedding in BMP Images," *Proceedings of 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, Kaohsiung, Taiwan, vol. 2, pp. 493-498, Nov. 2007

[8] P. H. W. Wong, O. C. Au and J. W. C. Wong, "A Data Hiding Technique in JPEG Compressed Domain," *Proceedings of 3rd SPIE International Conference on Security and Watermarking of Multimedia Contents*, San Jose, CA, USA, vol. 4314, pp. 309-320, Jan. 2001.

[9] C. H. Tzeng, Z. F. Yang and W. H. Tsai, "Adaptive data hiding in palette images by color ordering and mapping with security protection," *IEEE Transactions on Communications*, vol. 52, no. 4, pp. 791-800, 2004.

[10] A. Shamir, "How to share a secret," *Communication of the ACM*, vol. 22, pp. 612-613, 1979.

[11] C. C. Lin and W. H. Tsai, "Secret image sharing with steganography and authentication," *Journal of Systems and Software*, vol. 73, pp. 405-414, 2004.

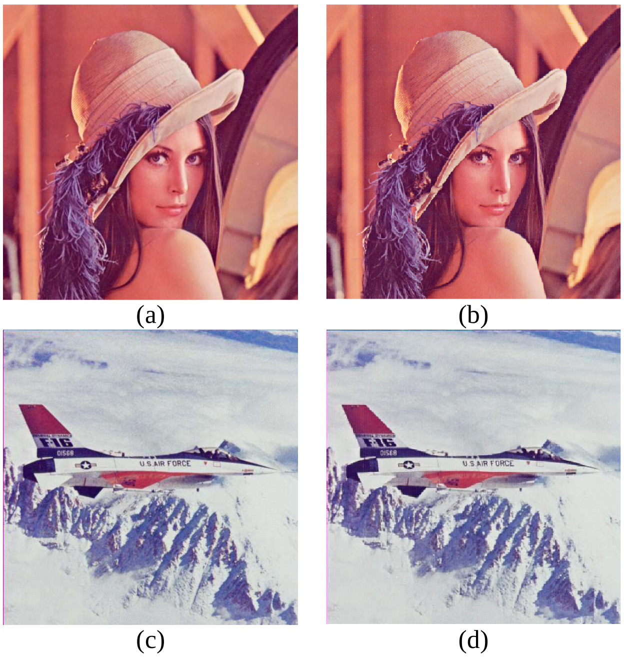
(a) (b) (c) (d)

Figure 1. Results of applying Algorithm 3 to embed a long sequence of binary message data into two images. (a) Cover image Lena. (b) Stego-image of Lena. (c) Cover image Jet. (d) Stego-image of Jet.

TABLE I. DATA HIDING CAPACITIES AND STEGO-IMAGE QUALITIES FOR T = 1 THROUGH 7 (DHC=DATA HIDING CAPACITY; PSNR=PEAK OF SIGNAL TO NOISE RATIO).

***t* value**	**Lena**		**Jet**	
	DHC (bits)	***PSNR of alpha channel (dB)***	***DHC (bits)***	***PSNR of alpha channel (dB)***
t = 1	262,144	45.44	262,144	45.44
t = 2	524,288	40.34	524,288	40.34
t = 3	786,432	32.69	786,432	32.69
t = 4	1,048,576	28.68	1,048,576	28.68
t = 5	1,310,720	21.72	1,310,720	21.72
t = 6	1,572,864	16.74	1,572,864	16.74
t = 7	1,835,008	10.61	1,835,008	10.61

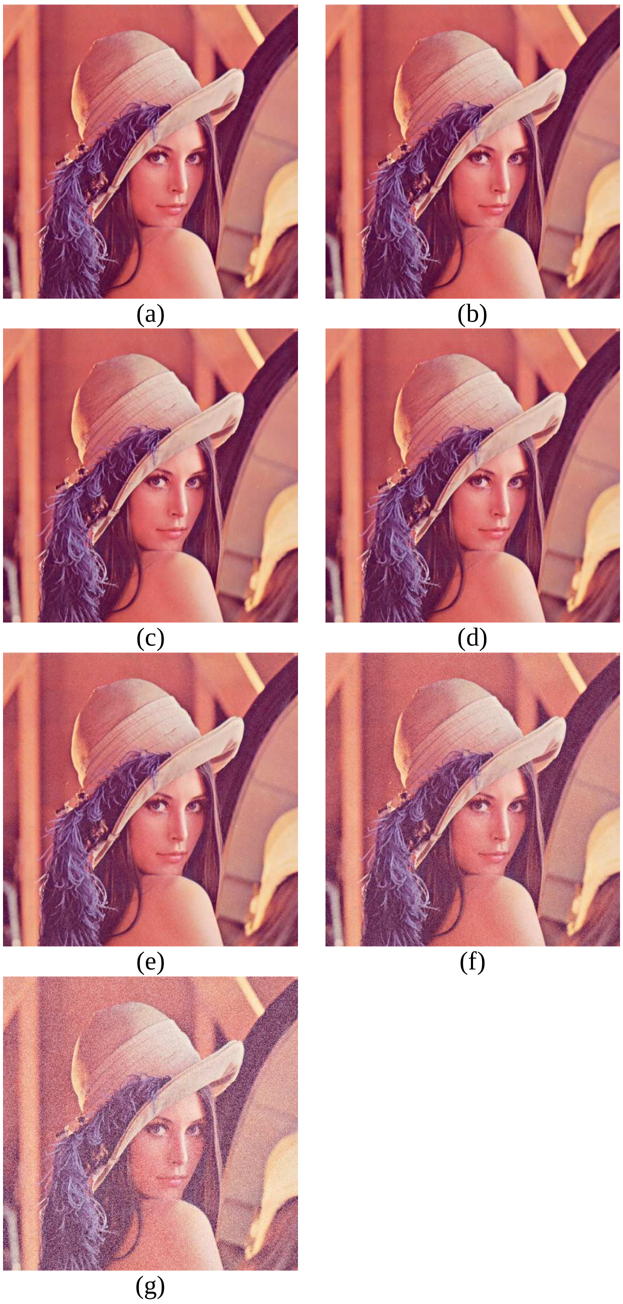
(a) (b) (c) (d) (e) (f) (g)

Figure 2. Stego-images yielded by Algorithm 3 for *t* = 1 through 7. (a) through (g) correspond to *t* = 1, 2, …, 7, respectively.