Bull. Soc. Photogr. Imag. Japan. (2016) Vol. 26 No. 1: 1–9

Original Paper

Hierarchical Scrambling Method for Palette-Based Images Using Bitwise Operation

Anu Aryal, Shoko Imaizumi, and Takahiko Horiuchi

Abstract: Image scrambling is a method in which an original image turns into a visually degraded image. In this paper, we propose a new algorithm of hierarchical scrambling method for palette-based images using bitwise operation. The target pixels for scrambling were chosen randomly using a pseudo-random number generator and were transformed into their corresponding binary bits. Then, bitwise operation was applied to concatenate the target pixels and the corresponding random numbers to scramble the palette-based images. The images in the proposed method are more distorted than those in the conventional methods. We also introduce a hierarchical key assignment scheme for more secure and efficient communication. We carried out performance tests using standard images. This demonstrates that our method is superior to the conventional methods.

Key words: Hierarchical scrambling, Palette-based images, Bitwise operation, Key assignment.

1. Introduction

The recent growth of the Internet has led to the arrival of various services that require sharing a huge amount of digital media and content through their websites. This widespread availability of digital media has created concerns about information leakage and privacy. There are various methods for information security that can be provided for digital media and communication.

Cryptographic techniques are used to scramble digital content so that it is extremely difficult for unauthorized users to decode it. Several kinds of scrambling methods are used for image protection. Full encryption [1], such as advanced encryption standard (AES) and data encryption standard (DES), is a robust way. However, it does not allow partial viewing and evaluation by users without the proper keys. Moreover, if there are any errors, it cannot decrypt the content correctly. Therefore, it is not always the best way to protect privacy and copyright. Digital watermarking [2] can be used for protecting copyright and hiding information in digital media such as images, videos, and audio content. It cannot, however, prevent unauthorized duplication because it does not visually degrade the quality of the image. Thus, watermarking also has limitations with regard to privacy protection. On another front, partial-scrambling schemes [3-11], where a secure algorithm is used to scramble only part of the medium, have been studied. Anyone can view the distorted images that are distributed by the providers, but only the authorized users can decrypt the images using their proper keys.

Palette-based images are generally limited to 256 colors. Hence, the amount of data is less than that of full-color images. Therefore, these images are used in many application areas such as multimedia, and websites. Furthermore, electronic paper (e-paper) has been very popular among general consumers lately [12]. The commercial success of monochrome e-paper led to the next generation of color e-paper. Some of the display companies provide not only monochrome e-paper, but also color e-paper. However, there is no full color e-paper in the current market so far. Moreover, the power consumption is larger as more colors need to be displayed [13].

A flexible partial encryption algorithm for palette-based images has been proposed [9]. However, the pixel scrambling method is not based on hierarchical access control. Therefore, quality degradation cannot be controlled depending on the access level granted. On the other hand, one conventional method [10] uses a basic operation that can only generate a few patterns. As a consequence, the method seems unsecure. In a similar way, another conventional method [11] involves scrambling the palette-based images, that is quite predictable and generates fewer patterns. As a result, these conventional methods are not secure enough to prevent malicious decryption by unauthorized users.

We propose a new hierarchical scrambling method for palette-based images using bitwise operation in this paper. In the proposed method, the target pixel threshold and the number of target color components (red (R), green (G), and blue (B)) are the main parameters for the quality control of scrambled images. The proposed method can degrade images more than the conventional methods [10][11] and also provides a secure environment against unau-

Received: 18th, December 2015; Accepted: 19th, May 2016
Graduate School of Advanced Integration Science, Chiba University
1-33 Yayoi-cho, Inage-ku, Chiba-shi, Chiba, 263-8522 Japan
Email: anu@chiba-u.jp, imaizumi@chiba-u.jp, horiuchi@faculty.chiba-u.jp

thorized decryption by using hierarchical key assignment schemes [14)15)].

The rest of this paper is organized as follows. In section 2, we present the technical specifications of our method. Then, we elaborate on the related works in section 3. In section 4, we describe the proposed hierarchical scrambling method in detail. The experimental results are demonstrated in section 5. We conclude this paper in section 6.

2. PRELIMINARY

Before discussing the proposed method, we describe some related technical terms. The colors in the palette are referred to as entities, and an index number is assigned to each entity. Each pixel in the image has an index number, and the index number indicates the corresponding entity, which specifies the RGB color.

2.1 Composition of Color Palette in Palette-Based Images

As shown in Fig. 1, each index number i of a color palette contains information specifying where the entity is located in the palette. The color palette itself is an array consisting of three columns corresponding to the primary colors R, G, and B, and N rows, where N is the number of entities in the palette. In the case of $N = 256$, the image is an 8-bit image.

2.2 Hierarchical Access Control

Access control is a technique to prevent unauthorized users accessing confidential information. Hierarchical key assignment schemes [14)15)] generate encryption keys so that the keys of a higher class can be used to derive the keys of all lower classes in the hierarchy. These schemes enable control of the quality level of the target information to be accessed.

Here, we assume that the scalable medium has two-dimensional scalability ($D = 2$). One of the dimensions describes the ratio of the pixels to be scrambled, and the other is the R, G, and B color components. Figure 2 depicts the hierarchical scrambling of two-dimensional scalability. The expected lowest quality image is R_{100} G_{100} B_{100}. The scrambling process is carried out on each entity with its corresponding key generated by a hierarchical key assignment scheme.

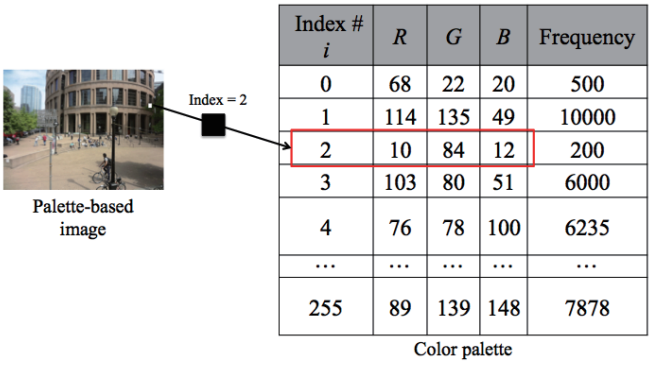


Fig. 1. Structure of palette-based image

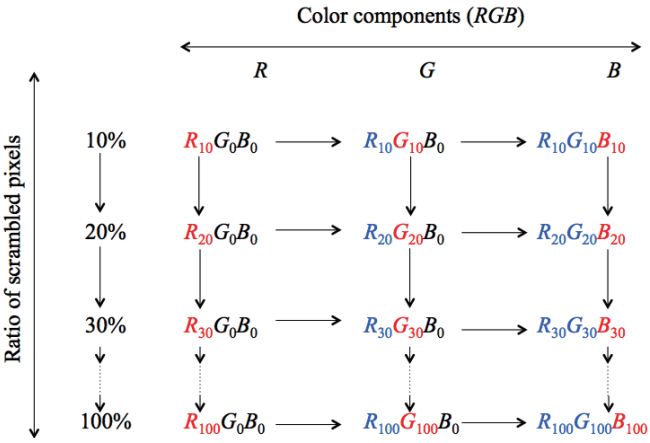


Fig. 2. Hierarchical scrambling of two-dimensional scalability ($D = 2$)[15)]

3. RELATED WORK

3.1 Hierarchical Scrambling of Palette-Based Images [10)]

One conventional method for hierarchical scrambling of palette-based images [10)] cyclically shifts the binary digits to scramble the images. The principal steps for this method are described as follows.

Step 1. Assume E_i is the i-th entity in the color palette.
Step 2. Calculate the frequency F_i of the i-th entity E_i.
Step 3. Sort all the entities E_i in ascending or descending order based on their frequencies F_i as shown in Fig. 3.
Step 4. Change the index number i of each entity E_i to j on the basis of the sorted color palette. Therefore, entities E_i are changed to E_j simultaneously.
Step 5. Select the target entities to be manipulated from the sorted palette in a hierarchical order.
Step 6. Extract an 8-bit binary value from R, G, or B values.
Step 7. Perform the cyclic shift operation on the binarized RGB values using a pseudo-random number.

For instance, as shown in Fig. 4, the decimal value '76' is chosen to be manipulated and is converted into the binary value equivalent of $(01001100)_2$. In the next step, a random number is chosen in the range 0 to 7 by using a pseudo-random number generator with its defined seed. In Fig. 4, the rightmost two digits are then cyclically shifted to the left, and the binary value becomes $(00010011)_2$, which is equivalent to '19'. In other words, this method performs cyclic shifts to manipulate the binary bits and scramble the images.

The major drawbacks of this approach are: 1) the cyclic shift operation is simple to perform and 2) we can change the current color in the palette to another color only from a limited range of colors due to the use of cyclic shift. Therefore, this method is vulnerable to malicious attacks, such as brute force attacks [16)]. Our main objective in this paper is proposing a more secure method.

3.2 Hierarchical Scrambling of Palette-Based Images Using Transposition Cipher [11)]

The basic idea of the conventional method [11)] is based on changing the positions of the R, G, and B. In Step 6 of *3.1*, the positions

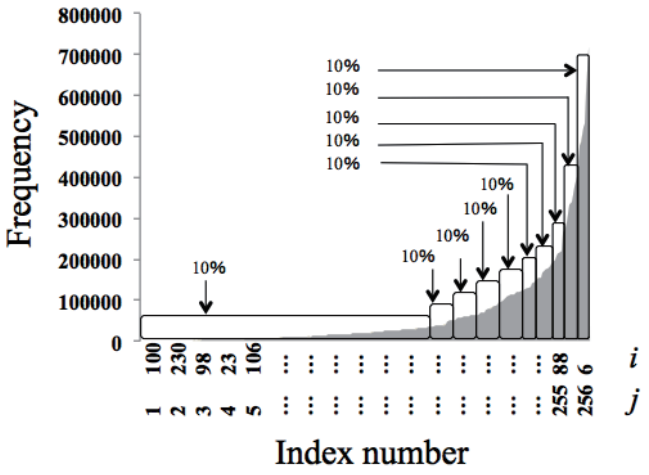


Fig. 3. Histogram sorted in ascending order

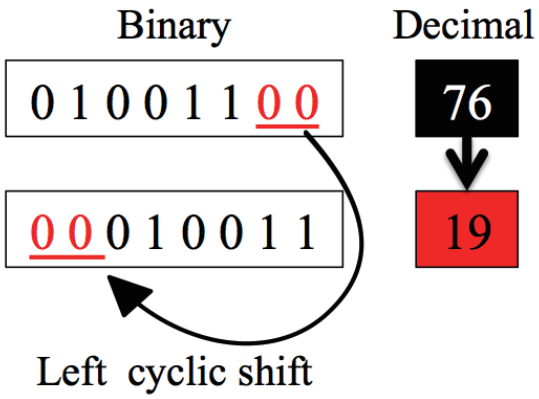


Fig. 4. Cyclic shift in conventional method [10]

of the *RGB* values of the target entities are randomly shuffled as shown in Fig. 5 to form the scrambled image. However, this scrambling process is too simple, and we can generate only a small number of patterns. Therefore, it is difficult to achieve sufficient complexity in the decryption process to thwart unauthorized users.

4. PROPOSED METHOD

In this section, we propose a hierarchical scrambling method for palette-based images using bitwise operation. The proposed method is superior to the conventional methods [10) 11)] in terms of the distortion of the scrambled images and security against malicious users.

Fig. 5. Procedure of conventional method [11]

4.1 Parameters for quality control

To control the quality of the scrambled images, we introduce two parameters: target pixels and target color components.

4.1.1 Target pixels: Figure 1 shows an example of the relation between the index number of entities and their frequency of use. Figure 3 shows a color palette sorted in ascending order and the process of selecting the frequencies of pixels in a hierarchical manner. The proposed method can control the quality of scrambled images and select scrambling from high-frequency regions when the color palette is sorted in ascending order or from low-frequency regions when the color palette is sorted in descending order.

4.1.2 Target color components: The process of manipulating the target *RGB* values is shown in Fig. 2. For instance, 10% of the *R*-component is first scrambled. The scrambled component is denoted as the R_{10} component. The scrambled image is then passed to the operation on the right to scramble 10% of the *G*-component, so the image is scrambled with the R_{10} and G_{10} components. In the next step, the result is again passed to the operation on the right for 10% scrambling of the *B*-component to obtain the R_{10}, G_{10}, and B_{10} components. In the case of 20% scrambling, the R_{10} component is passed to the operation below for the next 10% scrambling to obtain the R_{20} scrambled image. The resultant R_{20} is then passed to the operation on the right, and the G_{10} component is passed to the operation below for the next 10% scrambling, so the image is scrambled with the R_{20} and G_{20} components. In the next step, the resultant R_{20} and G_{20} are passed to the operation on the right, and B_{10} is passed to the operation below for the next 10% scrambling to obtain the scrambled image with R_{20}, G_{20}, and B_{20} components. This process is repeated for all pixels.

4.2 Algorithm

Fig. 6 shows the outline of the proposed method. We describe the scrambling procedure of our method in detail.

Step 1. Assume E_i is the *i*-th entity in the color palette.

Step 2. Calculate the frequency F_i of the *i*-th entity E_i.

Step 3. Sort all the entities E_i in ascending or descending order based on their frequencies F_i as shown in Fig. 3.

Step 4. Change the index number *i* of each entity E_i to *j* on the basis of the sorted color palette. Therefore, entities E_i are changed to E_j simultaneously.

Step 5. Select the target entities to be manipulated from the sorted palette in a hierarchical order.

Step 6. Extract an 8-bit binary value from *R*, *G*, or *B* values.

Step 7. Choose a random number and perform modulo operation with respect to *N*, where *N* is 256. The decimal result is then converted into its binary value as shown in Fig. 6.

Step 8. Apply bitwise operation to the binary results from Steps 6 and 7.

Step 9. Repeat Steps 5-8 until all of the target entities and the target components are modified.

Steps 1 to 6 are carried out as described above in section *3.1* for this

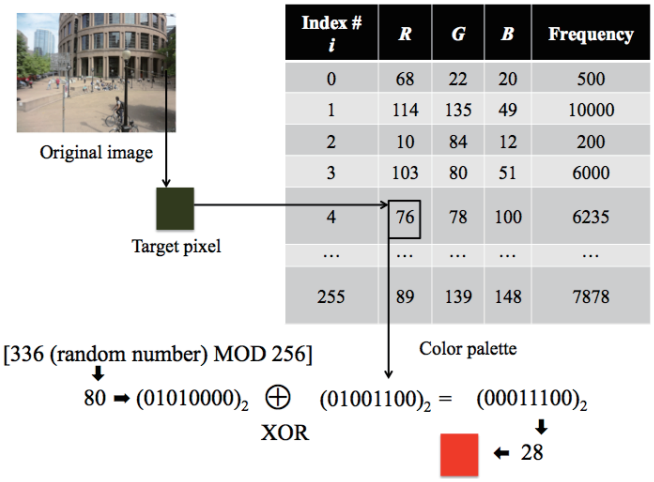


Fig. 6. Overview of proposed method

method. Note that we adopt XOR operation as the bitwise operation in this paper.

4.3 Key assignment and Encryption

We have adopted hierarchical key assignment schemes using simple hash chains (SHCs) and recursive hash chains (RHCs)[15], as depicted in Fig. 7. Basically, each scrambling class is assigned a key that can generate the keys of all lower classes in the hierarchy.

Access control for scalable media should encrypt the codestream entity-by-entity by using different keys. Keys for encryption are derived as shown in Fig. 7. For example, keys K_{R10}, K_{G10}, and K_{B10} are the keys for the $R10$, $G10$, and $B10$ components, respectively. K_{R10} is the managed key. K_{R20} is derived from K_{R10} as follows:

$$K_{R20} = H(K_{R10}), \tag{1}$$

where $H(\cdot)$ is a cryptographic one-way hash function, i.e., SHA-256 [17]. Similarly, the other keys K_{Rs} (s = 30, 40, …, 100) are derived with SHCs.

On the other hand, keys K_{G10} and K_{B10} are derived by RHCs as follows:

$$\begin{aligned} K_{G10} &= H(f(K_{R10}, H(K_{R10}))) \\ &= H(f(K_{R10}, K_{R20})), \end{aligned} \tag{2}$$

and

$$\begin{aligned} K_{B10} &= H(f(K_{G10}, H(K_{G10}))) \\ &= H(f(K_{G10}, K_{G20})), \end{aligned} \tag{3}$$

where $f(\cdot)$ is a function that represents bitwise XOR operation between the two inputs.

In this case, multiple keys can be recursively computed from the master key by using hash chains. The main benefits of using hash chains are: 1) they can reduce the complication of key management and delivery and 2) they can be independent from each other, which

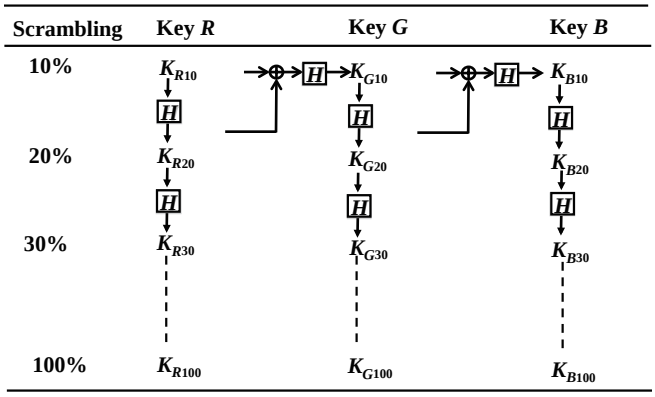


Fig. 7. Key assignment using SHCs and RHCs

make the security robust against collusion attacks. These keys are the seeds of pseudo-random numbers in this context. Moreover, the key management of keeping only a single key K_{R10} is quite simple. Note that the decryption procedure requires both the pseudo-random numbers and the keys.

4.4 Key delivery and decryption

As shown in Fig. 8, if the user is allowed to access the image with the quality of $R_{20}G_{20}B_{20}$, the user receives the three keys K_{R30}, K_{G30}, and K_{B30}. The 21 keys K_{R40}, K_{G40}, …, K_{B100} are derived from the three delivered keys by using SHCs. The user can decrypt the corresponding 24 components such as the $R30$, $G30$, …, $B100$ components.

Similarly, as shown in Fig. 9, if the user is allowed to access the image with the quality of $R_{100}G_{100}B_0$, the user receives key K_{B10}. The 9 keys are derived from the delivered key using SHCs. The user can decrypt the corresponding 10 components such as $B10$, $B20$, …, $B100$ components.

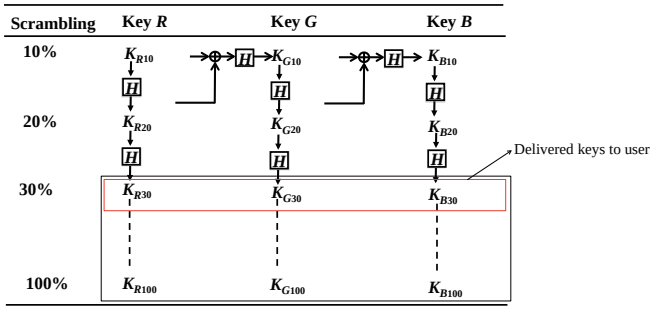


Fig. 8. Decryption process for image quality of $R_{20}G_{20}B_{20}$

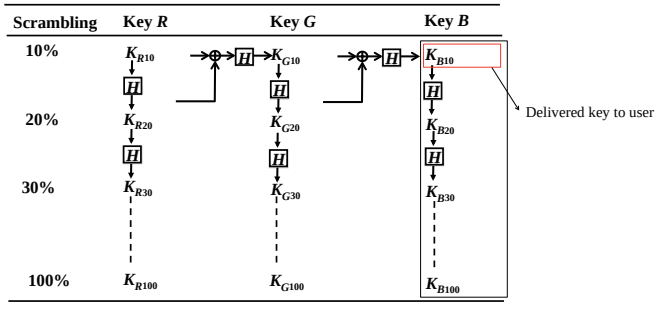
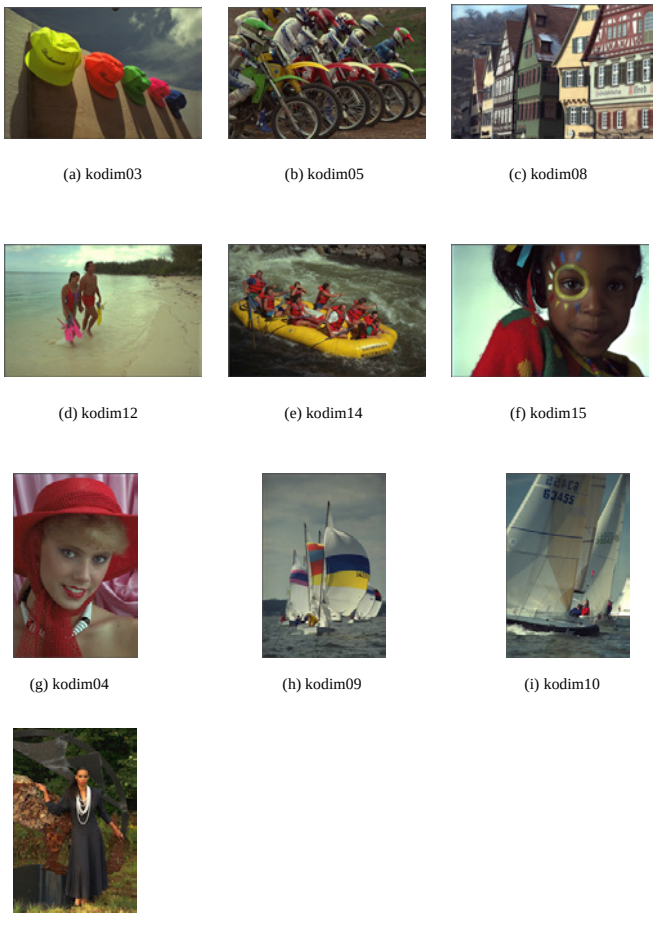


Fig. 9. Decryption process for image quality of $R_{100}G_{100}B_0$

5. EXPERIMENTAL RESULTS

The proposed and conventional methods [10)11)] were examined by using ten different images available from the Kodak lossless true color image suite [18)] as shown in Fig. 10. The original images are of two different sizes: 512 × 768 and 768 ×512. The private text included in the images, such as the number plate of an automobile, is an important factor to consider for security. Therefore, we utilized some of the original images such as kodim03, kodim08, kodim09, kodim10, and kodim14 to check if the text in those images can be recognized after scrambling. Similarly, as it is important to evaluate the dominant ratio of human faces in the image, we considered images with small (kodim12 and 14), medium (kodim 18), and large faces (kodim04 and 15) as shown in Fig. 10 for the simulation.

For the simulation, the colors of the 24-bit images were reduced before scrambling. The minimum variance quantization was performed on the original images by calling matlab function 'rgb2ind', specifying the maximum number of colors in the output as 32 (5-bit), 64 (6-bit), 128 (7-bit), and 256 (8-bit) colors without setting dithering option. We carried out a performance evaluation with the structural similarity (SSIM) index [19)], peak signal-to-noise ratio (PSNR), and mean square error (MSE) between the original image and the visually distorted images for the proposed method as well as the conventional methods. The parameters used while calculating



(a) kodim03 (b) kodim05 (c) kodim08

(d) kodim12 (e) kodim14 (f) kodim15

(g) kodim04 (h) kodim09 (i) kodim10

(j) kodim18

Fig. 10. Original images from Kodak lossless true color image suite [18)]

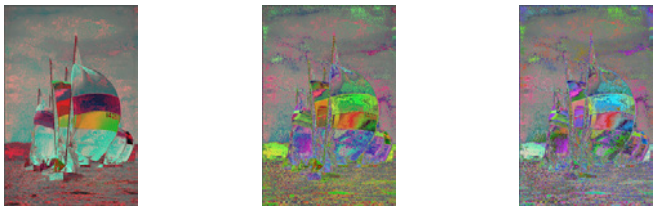
the SSIM values were defined as K (constants in the SSIM index formula) = [0.01, 0.03], *Window* (local window value) = 11, and L (dynamic range of image) = 1.

5.1 Analysis of Image quality

Figure 11 shows 30%, 60%, and 100% scrambling of the original image 'kodim09' with the maximum number of colors as 256. The quality of the scrambled images is gradually distorted with increasing percentage of target pixels. Furthermore, note that the text included in the original image 'kodim09' is also gradually distorted and is difficult to read as the percentage of scrambling is increased. Figure 12 shows the three scrambled images of 'kodim09' with 32 colors, where the quality is $R_{100}G_{100}B_{100}$, using different seeds for the pseudo-random generator. Figure 12 (a), (b), and (c) look different from each other. For example, the text in the image in Fig. 12 (a) has been distorted the most as we cannot recognize the text. On the other hand, Fig. 12 (c) has been distorted less as we can easily read the text. Therefore, the quality of the scrambled images can be controlled by using different seeds. Figure 13 shows the monochrome version of the scrambled images of image 'kodim09' with the intensity level as 32. The proposed method is applied on 'image09' after converting to monochrome version. It is seen that the quality of the scrambled image is degraded hierarchically for the monochrome image also. Figure 14 shows the full scrambled images of 'kodim09' (monochrome) with 32 intensity levels, using three different seeds. As shown in the simulations, it is difficult to see whether people are present in the ships or not. In addition, the text written on the flag of the ship, which is present at the back, are distorted completely and are invisible. On the other hand, the upper text written on the flag of ship, which is present at the front is visible, whereas the lower text is considerably degraded and is difficult to read the numbers.

Figure 15 shows 30%, 60%, and 100% scrambling of the original image 'kodim18' with the maximum number of colors as 256. In the case of 100%, the image is distorted the most. Note that the color of the woman's skin and the background, where she is standing slowly become distorted with increasing percentage of target pixels. At the quality of $R_{100}G_{100}B_{100}$ scrambling, it is hard to see the actual color of her skin and background of image. From the simulation results, we can also observe that the visual quality is degraded as the number of target color components is increased. In general, the simulation results show that our hierarchical scrambling scheme is effective for these test configurations. In other words, the image quality is degraded in a hierarchical manner as the percentage of target pixels is increased. Figure 16 shows the three scrambled images of 'kodim18' with 32 colors, where the quality is $R_{100}G_{100}B_{100}$, using different seeds. Figure 16 (a), (b), and (c) are different. It is difficult to see the actual color of woman's necklace in all of the figures, but the outline of her necklace is visible and cannot be hidden. Figure 17 represents the monochrome version of the scrambled images of original image 'kodim18' with 32 intensity levels. Figure 18 shows the full scrambled images of 'kodim18' (monochrome) with 32 intensity levels, using three different seeds. As shown in the simulation results, the face of the woman is distorted and is difficult to see.

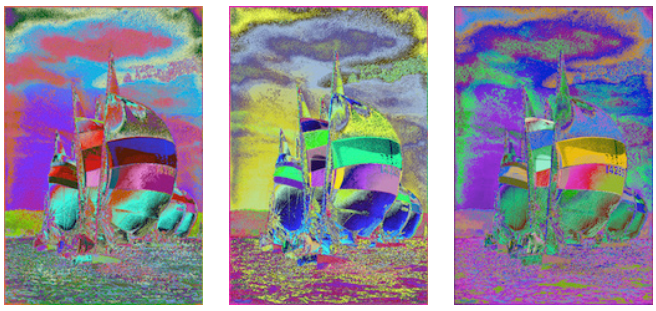
(a) $R_{30}G_0B_0$ (b) $R_{30}G_{30}B_0$ (c) $R_{30}G_{30}B_{30}$



(d) $R_{60}G_0B_0$ (e) $R_{60}G_{60}B_0$ (f) $R_{60}G_{60}B_{60}$

(g) $R_{100}G_0B_0$ (h) $R_{100}G_{100}B_0$ (i) $R_{100}G_{100}B_{100}$

Fig. 11. Scrambled images in proposed method ('kodim09' with 256 colors)



(a) Seeding pattern A (b) Seeding pattern B (c) Seeding pattern C

Fig. 12. Full scrambled images ($R_{100}G_{100}B_{100}$) of 'kodim09' with 32 colors in proposed method using three different seeds

5.2 Analysis of SSIM, PSNR, and MSE metric

We evaluated the quantitative performance of the proposed scheme by calculating the SSIM, PSNR, and MSE values of all the distorted images. Figure 19 shows the SSIM results of 'kodim18' that were obtained by scrambling the *R*, *G*, and *B* components where the maximum numbers of colors were 32, 64, 128, and 256. We controlled two parameters: the target pixels and the target color components in the experiment. The target pixels were chosen from the target color components in a hierarchical manner. We realize that there is a hierarchical scrambling of images for all quantized colors. Figure 20 and 21 show the PSNR and MSE values of 'kodim18' obtained by scrambling the *R*, *G*, and *B* components for different numbers of colors. We can observe that the value of PSNR decreases as the percentage of scrambling increases. The value of MSE increases as the percentage of scrambling increases because the higher

(a) 10% (b) 20% (c) 30%

(d) 40% (e) 50% (f) 60%

(g) 70% (h) 80% (i) 90%

(j) 100%

Fig. 13. Scrambled images in proposed method ('kodim09' with 32 intensity levels, monochrome)

(a) Seeding pattern A (b) Seeding pattern B (c) Seeding pattern C

Fig. 14. Full scrambled images of 'kodim09' (monochrome) with 32 intensity levels in proposed method using three different seeds

the value of MSE, the higher the distortion level is. Figure 22, 23, and 24 show the results of SSIM, PSNR, and MSE, respectively, of 'kodim18' with 256 colors by scrambling (*R*), (*R* and *G*), and (*R*, *G*, and *B*) components. These graphs show that the scrambling process occurred in a hierarchical manner.

Table 1 shows the SSIM, PSNR, and MSE values for 'kodim18'

(a) $R_{30}G_0B_0$ (b) $R_{30}G_{30}B_0$ (c) $R_{30}G_{30}B_{30}$

(d) $R_{60}G_0B_0$ (e) $R_{60}G_{60}B_0$ (f) $R_{60}G_{60}B_{60}$

(g) $R_{100}G_0B_0$ (h) $R_{100}G_{100}B_0$ (i) $R_{100}G_{100}B_{100}$

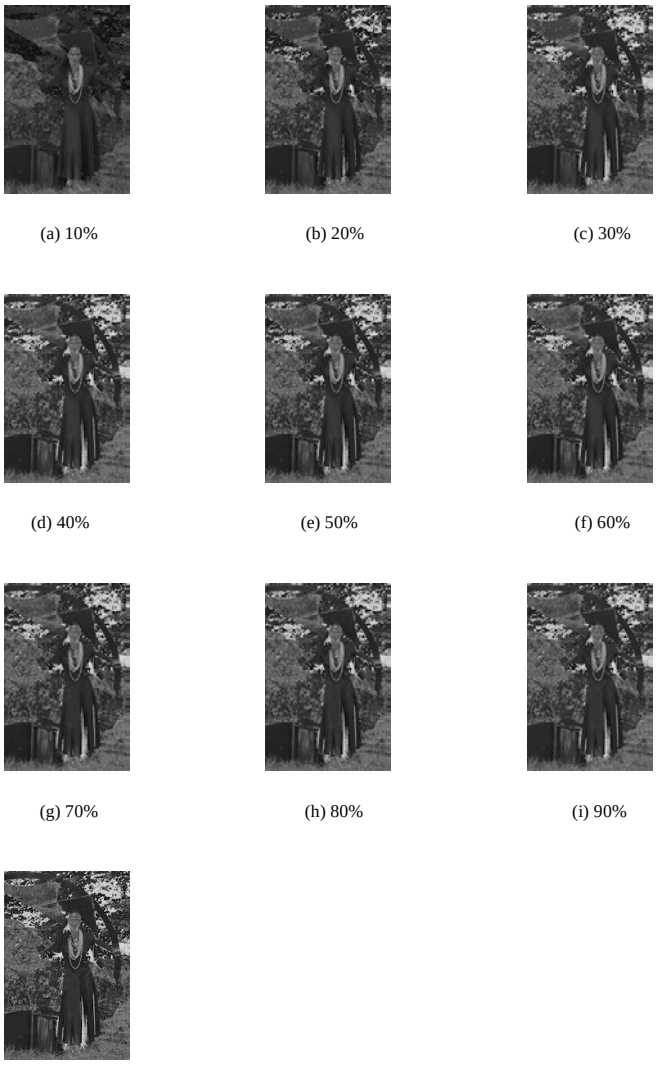
Fig. 15. Scrambled images in proposed method ('kodim18' with 256 colors)

(a) Seeding pattern A (b) Seeding pattern B (c) Seeding pattern C

Fig. 16. Full scrambled images ($R_{100}G_{100}B_{100}$) of 'kodim18' with 32 colors in proposed method using three different seeds

with 256 colors where the target color components are the *R*, *G*, and *B* components. The values of SSIM in the proposed method are lower compared with those of the two conventional methods [10)11)]. The proposed method can distort the images more than the two conventional methods. Furthermore, the proposed method has the most degraded image because the value of SSIM for 100% scrambling in the proposed method is much lower, i.e. 0.006, than that of other two methods, which have SSIM values of 0.082 and 0.311. Similarly, the conventional methods have larger values of PSNR than the proposed method. The MSE values in the proposed method are higher than those of the conventional methods. Therefore, the images are more distorted in the proposed method than in the conventional methods [10)11)].

The proposed method is quite efficient compared to the above-mentioned methods [10)11)] because in the proposed method we



(a) 10% (b) 20% (c) 30%

(d) 40% (e) 50% (f) 60%

(g) 70% (h) 80% (i) 90%

(j) 100%

Fig. 17. Scrambled images in proposed method ('kodim18' with 32 intensity levels, monochrome)

(a) Seeding pattern A (b) Seeding pattern B (c) Seeding pattern C

Fig. 18. Full scrambled images of 'kodim18' (monochrome) with 32 intensity levels in proposed method using three different seeds

can change the current color in the palette to another one from the 24-bit colors, that is, full colors. The conventional method [10)] has a cyclic shift operation to manipulate the binary bits by choosing a random number in the range 0 to 7. Therefore, we can change the current color in the palette to another one only from the limited colors but not from the full colors. Similarly, the conventional method [11)] shuffled the positions of the *RGB* values of target entities ran-

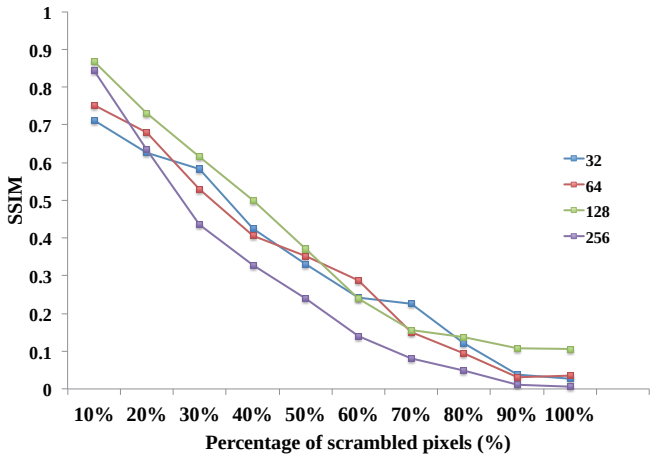


Fig. 19. SSIM values of 'kodim18' obtained by scrambling *R*, *G*, and *B* components for different numbers of colors

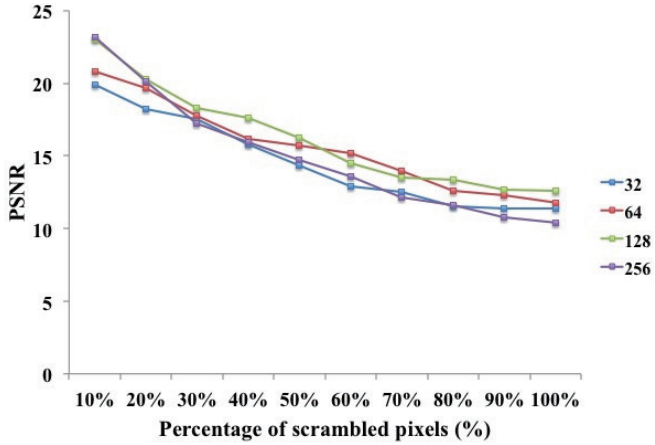


Fig. 20. PSNR values of 'kodim18' obtained by scrambling *R*, *G*, and *B* components for different numbers of colors

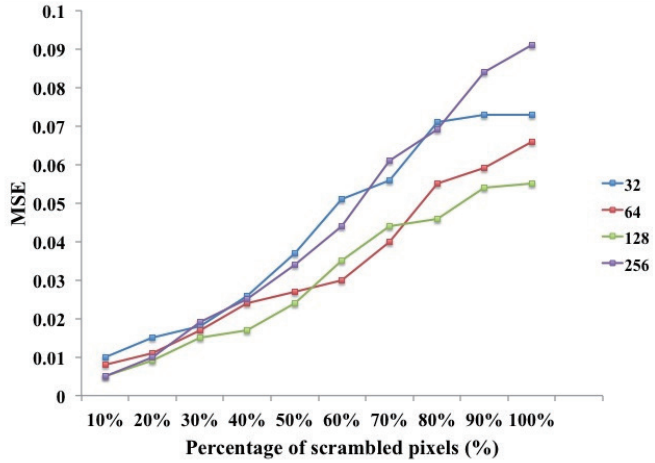


Fig. 21. MSE values of 'kodim18' obtained by scrambling *R*, *G*, and *B* components for different numbers of colors

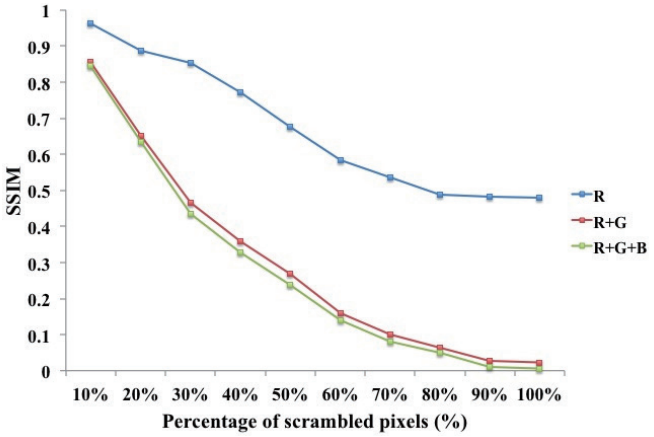


Fig. 22. SSIM values of 'kodim18' with 256 colors obtained by scrambling (*R*), (*R* and *G*), and (*R*, *G*, and *B*) components

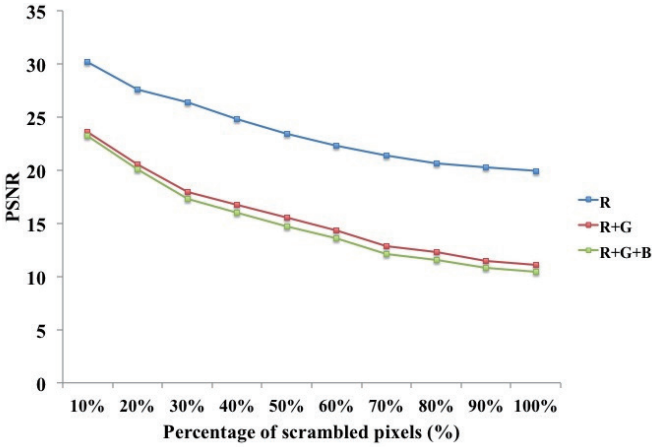


Fig. 23. PSNR values of 'kodim18' with 256 colors obtained by scrambling (*R*), (*R* and *G*), and (*R*, *G*, and *B*) components

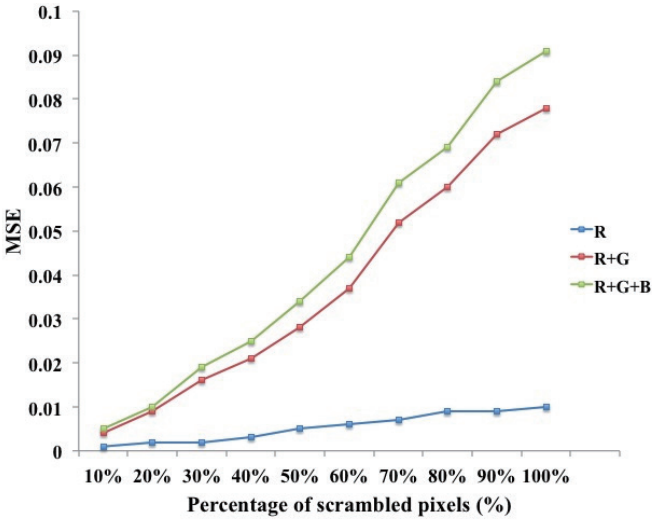


Fig. 24. MSE values of 'kodim18' with 256 colors obtained by scrambling (*R*), (*R* and *G*), and (*R*, *G*, and *B*) components

domly in the range 1 to 256. Hence, we cannot change the colors to one from the full colors.

6. CONCLUSION

In this paper, we have proposed a new algorithm for a hierarchical scrambling method for palette-based images using bitwise operation. The proposed method distorts the images more than the conventional methods [10)11)] and also enhances the security of scrambling for palette-based images. The colors in an image can be changed to any color, from the 24-bit colors in our proposed method, whereas

Table 1. SSIM, PSNR, and MSE values of 'kodim18' with 256 colors obtained by scrambling *R*, *G*, and *B* components

Percentage of scrambled pixels (%)	SSIM			PSNR			MSE		
	Prop.	Conv. [10]	Conv. [11]	Prop.	Conv. [10]	Conv. [11]	Prop.	Conv. [10]	Conv. [11]
10%	0.844	0.915	0.861	23.192	26.072	24.658	0.005	0.002	0.003
20%	0.635	0.683	0.749	20.088	20.978	22.659	0.010	0.008	0.005
30%	0.436	0.537	0.697	17.267	19.360	21.985	0.019	0.012	0.006
40%	0.327	0.386	0.573	15.962	16.999	20.303	0.025	0.020	0.009
50%	0.239	0.318	0.509	14.697	15.913	19.855	0.034	0.026	0.010
60%	0.139	0.204	0.475	13.573	14.504	19.721	0.044	0.035	0.011
70%	0.082	0.160	0.413	12.149	13.612	19.332	0.061	0.044	0.012
80%	0.049	0.126	0.354	11.588	13.169	19.123	0.069	0.048	0.012
90%	0.010	0.085	0.313	10.769	12.465	18.914	0.084	0.057	0.013
100%	0.006	0.082	0.311	10.407	12.220	18.880	0.091	0.060	0.013

there is a restriction on changing colors from the full colors in the conventional methods. Therefore, the proposed algorithm is more secure against unauthorized decryption compared to the conventional methods. Moreover, it can control the quality of scrambled images using only a single managed key. Our future work involves consideration of additional parameters to develop a more robust platform for effective sharing and controlling of the target images.

Acknowledgement

This work was supported by Venture Business Laboratory (VBL) Research Project of Chiba University.

References

1) I. E. Ziedan, M. M. Fouad, and D. H. Salem, "Application of data encryption standard to bitmap and JPEG images," in *Proc. IEEE National Radio Science Conference*, pp. C16: 1–8, 2003.
2) V. M. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques," in *Proc. IEEE International Conference on Industrial Informatics*, pp. 709–716, 2005.
3) H. Kiya, S. Imaizumi, and O. Watanabe, "Partial-scrambling of images encoded using JPEG2000 without generating marker codes," in *Proc. International Conference on Image Processing*, vol. 2, pp. III205–III208, 2003.
4) J. M. Rodrigues, W. Puech, and A. G. Bors, "Selective encryption of human skin in JPEG images," in *Proc. IEEE International Conference on Image Processing*, pp. 1981–1984, 2006.
5) A. K. Yekkala, N. Udupa, N. Bussa, and C. V. Madhavan, "Lightweight encryption for images," in *Proc. International Conference on Consumer Electronics*, pp. 1–2, 2007.
6) P. -C. Su, W. -Y. Chen, S. -Y. Shiau, C. -Y. Wu, and A. Su, "A privacy protection scheme in h. 264/avc by data hiding," in *Proc. Asia-Pacific Signal and Information Processing Association*, pp. 1–7, 2013.
7) A. Said, "Measuring the strength of partial encryption schemes," in *Proc. IEEE International Conference on Image Processing*, vol. 2, pp. II: 1126–9, 2005.
8) M. Fujiyoshi, S. Imaizumi, and K. Hitoshi, "Encryption of composite multimedia contents for access control," *IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E90–A, no. 3, pp. 590–596, 2007.
9) S. Imaizumi, and Y. Tobe, "Flexible partial encryption for palette-based images," in *Proc. International Workshop on Advanced Image Technology*, no. 160 (CD–ROM), 2015.
10) A. Aryal, S. Imaizumi, and N. Aoki, "Hierarchical scrambling scheme for palette-based images," in *Proc. IEEE International Symposium on Intelligent Signal Processing and Communication Systems*, no. 74, pp. 65–70, 2014.
11) A. Aryal, S. Imaizumi, and N. Aoki, " Hierarchical scrambling for palette-based images using transposition cipher," in *Proc. of International conference on Advanced Imaging*, no. T109–01, pp. 701–704, 2015.
12) J. Heikenfeld, P. Drzaic, J. S. Yeo, and T. Koch, " A critical review of the present and future prospects for electronic paper," *J. Society for Information Display*, vol. 19, no. 2, pp. 129–156, 2011.
13) R. Kubota, H. Tamukoh, H. Kawano, N. Suetake, B. Cha, and T. Aso, "A color quantization based on vector error diffusion and particle swarm optimization considering human visibility," in *Proc. of 7th Pacific-Rim Symposium on Image and Video Technology*, pp. 332–343, 2015.
14) H. -F. Huang, and C. -C. Chang, "A new cryptographic key assignment scheme with time-constraint access control in a hierarchy," *Computer Standards & Interfaces*, vol. 26, no. 3, pp. 159–166, 2004.
15) S. Imaizumi, "A collusion-free key assignment scheme for hierarchical access control using recursive hash chains," in *Proc. IEEE International Symposium on Circuits and System*, pp. 445–448, 2013.
16) L. R. Knudsen, and M. J. Robshaw, "Brute Force Attacks," *The Block Cipher Companion. Springer-Verlag*, pp. 95–108, 2011.
17) NIST, Secure Hash Standard, FIPS PUB 180–4, 2012
18) Kodak Lossless True Color Image Suite, http://r0k.us/graphics/kodak/.
19) Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.