A study on hierarchical protection for copyrights of digital images

July 2018

ANU ARYAL
Graduate School of
Advanced Integration Science
CHIBA UNIVERSITY

(千葉大学審査学位論文)

A study on hierarchical protection for copyrights of digital images

July 2018

ANU ARYAL
Graduate School of
Advanced Integration Science
CHIBA UNIVERSITY

Acknowledgements

I would like to express my heartfelt thanks to my supervisor, Prof. Takahiko Horiuchi for his creative guidance, stimulating discussions, and valuable suggestions. Similarly, I owe my deepest gratitude to my research advisor Assoc. Prof. Shoko Imaizumi, whose intellectual support, valuable advice, inspiring words, and critical comments enabled me to become familiar with my research. It would not have been possible to finish this work without her support and motivation. I would also like to acknowledge Prof. Yoshitsugu Manabe and Prof. Toshiya Nakaguchi for providing their valuable feedback on my dissertation. Without their suggestions, the dissertation could not have been successfully completed. I also express my profound gratitude to Prof. Hitoshi Kiya for his continuous advice throughout my doctoral study.

I am indebted to my family for their trust that made me confident and strong enough to accomplish my work. I would like to thank my parents, sisters, and brothers for their love and prayers. Most importantly, I wish to thank my wonderful husband as well as the son for their support and encouragement.

Finally, I express my sincere appreciation to all friends, supervisors, administrators, and staffs of Chiba University for their help and support.

Executive Summary

While the rapid growth of information technology has led to the wide use of various Internet services, the security of digital multimedia is very important to prevent malicious distribution from unauthorized users. Image encryption is one of the image protection methods. The author focuses on protecting images hierarchically and provides access control for different users. In this dissertation, the author first proposes a new algorithm for a hierarchical scrambling method for palette-based images using bitwise operation. The target pixel values for scrambling are taken by using pseudo-random numbers. Then, bitwise exclusive-OR operations are applied to concatenate the target pixel values and the corresponding pseudo-random numbers to manipulate the original pixel values. The author introduces a hierarchical key assignment scheme to control the various access rights. Secondly, the author proposes an integrated model of Block-Permutation-Based Encryption (BPBE) and reversible data hiding. This method allows a hierarchical process for encryption and embedding. Hence, my method can be suitable for the hierarchical access control system, where the permission is allowed according to different access rights. In addition, the key derivation scheme provides the security according to the different access rights. This method is also compatible with the standard lossless compression methods such as JPEG-LS as the compression performance is not severely degraded. Furthermore, this method is resilient against brute-force attacks as well as jigsaw puzzle solvers. The author implemented the proposed algorithms and verified the feasibility of the methods through experiments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The recent advancement in information and communication technology has led to the enormous distribution of digital multimedia content through the Internet. With the availability of Internet access, there are various services such as Social Networking Services (SNSs), cloud services, and digital archiving services. Therefore, security of the digital properties has become highly important since they can be easily duplicated and manipulated over the open network. There are different methods for information security on computers and over the Internet. One of the most popular forms of security is encryption. Encryption is the conversion of a plain text into a cipher text, which cannot be read without decrypting the text. Decryption is the reverse process of converting an encrypted text into an original plain text. The system of encryption and decryption process is referred to as a cryptosystem. The security of digital images, which are transmitted over an open network, can be achieved with image encryption, where only authorized users are able to decrypt the original image.

Generally, there exist different approaches of image security methods. Encryption and information hiding are the popular methods, which can be broadly classified as given in Fig. 1.1. Watermarking [1, 2] is the process of embedding information into an image to protect the ownership of the image. It can be used to hide copyright information into digital media and protect it. However, it has limitations on unauthorized duplications as it does not severely degrade the image quality. In image encryption, the quality of the images is degraded. The authorized users can only be provided with a proper key to decrypt the original

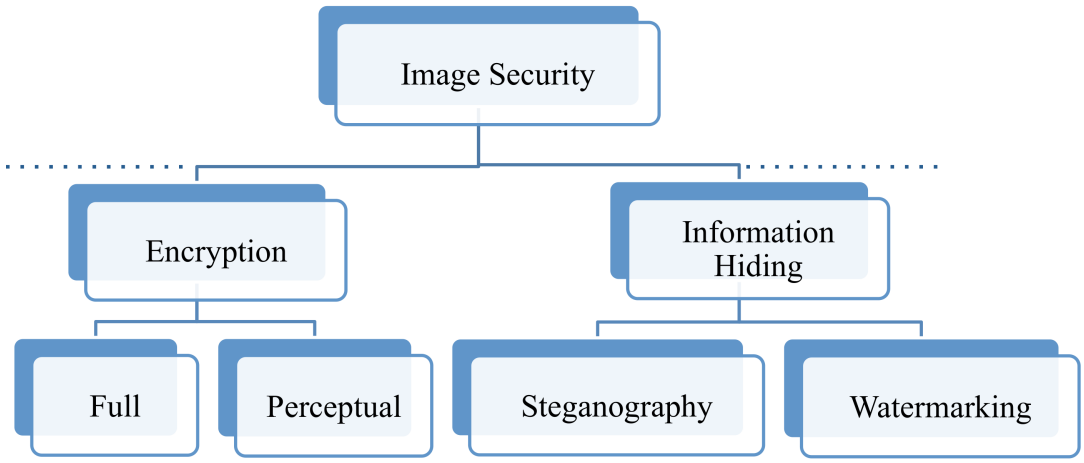


Figure 1.1: Image security methods.

image. Full encryption [3, 4] such as Advanced Encryption Standard (AES) and triple Data Encryption Standard (DES), is an efficient way to encrypt the images. However, there is a trade-off between computational complexity and security strength for the various requirements such as low processing demands, signal processing in an encrypted domain, and so on. In full encryption, the users are not allowed to have a partial view without obtaining the proper keys. To consider this issue, perceptual encryption schemes [5–22] has been studied. Perceptual encryption is an operation that makes images difficult to understand visually. However, the encrypted image can be viewed and analyzed by any users, but the proper key is only provided to the authorized users to decrypt the image. This method also reduces the encryption and the decryption costs.

A hierarchical access control system allows different access rights for users at various permission levels. Hence, it prevents unauthorized users from accessing the confidential information. The hierarchical key assignment schemes [19–25] provide the proper encryption keys to each user with different access rights in the hierarchy. As these encryption keys should be independent, a large amount of keys would be generated. Therefore, the proper key management and the delivery of those multiple keys can be more complicated. The author considers to derive an efficient key management scheme with the use of hash chains [24] and decrease the number of managed keys. With the use of hash chains, the derived keys can

be computed from a single managed key. By using hash chains, the keys can be independent from each other and make the security robust against collusion attacks.

In the first half of this dissertation, the author focuses on developing a hierarchical scrambling method for palette-based images. Palette-based images are 8-bit images and use no more than 256 colors. Therefore, these images can be used in different application areas such as websites and multimedia. Nowadays, an electronic paper (e-paper) has been very popular [26]. Some of the display companies provide monochrome as well as color e-paper. However, the full color e-paper still does not exist in the current market. The power consumption for displaying full colors is larger because more colors need to be displayed [27]. Hence, the palette-based images are important as they consume less power to display the limited colors. Considering the importance of encryption in palette-based images, a flexible partial encryption for them has been proposed [10]. However, there is no concept of the hierarchical access control while scrambling the images in this scheme [10]. This means that it is not possible to control the distortion of images according to various access levels. Similarly, the conventional method [19] uses cyclic shift operations, which can only generate a few patterns, and thus the method is not secure. Another partial scrambling method [20] was proposed, which is quite predictable and generates fewer patterns. Therefore, these methods do not provide proper security to prevent malicious attacks by unauthorized users. Thus, the main objective of this dissertation is to develop a new hierarchical scrambling method for palette-based images using bitwise operation. The target pixel values for scrambling are taken by using pseudo-random numbers. Then, bitwise exclusive-OR operations are employed to concatenate the target pixel values and the corresponding pseudo-random numbers to manipulate the original pixel values. The author introduces a hierarchical key assignment scheme to control the various access rights.

In the second half of this dissertation, the objective is to utilize data hiding as well as encryption in a hierarchical manner. There are two types of data hiding methods, namely, Irreversible Data Hiding (IDH) [28–30] and Reversible Data Hiding (RDH) [31–46]. While the host image cannot be completely recovered in IDH, the host image should be completely recovered in RDH. Hence, RDH

techniques are desirable in different areas, such as medical imagery, military communications, and law forensics, where no permanent change is permitted. In this dissertation, the author utilizes RDH to hide the target information. On the other hand, the Block-Permutation-Based Encryption (BPBE) schemes [47–50] are one of the perceptual encryption techniques that has been proposed for Encryption-then-Compression (ETC) technique [51–57]. Consequently, the main motivation is to integrate BPBE and RDH for the hierarchical access control for the encryption and the data embedding processes. The BPBE scheme guarantees confidentiality, while the RDH allows the target information to be hidden into an image. The proposed method introducing BPBE also considers to maintain the compression efficiency by using international standards for image compression. This method can be attractive in the scenario, such as doctor-nurse in a hospital, large organizations, and hierarchical file systems, where there is a hierarchical access control according to the various access rights.

The succeeding chapters of this dissertation are organized as follows. Chapter 2 includes the technical requirements. The hierarchical access control system is elaborated in Section 2.1. Section 2.2 gives a brief overview of digital image formats, such as BMP, GIF, PNG, TIFF, EPS, and JPEG. Section 2.3 provides the mathematical expression to calculate the different qualitative metrics. Section 2.4 describes RDH. Section 2.5 elaborates BPBE scheme. Section 2.6 discusses cryptographic hash function and its security properties. The compression techniques are described in Section 2.7.

Chapter 3 proposes a new algorithm of a hierarchical scrambling method for palette-based images using bitwise operation [21]. Section 3.1 discusses the technical preliminaries such as a composition of color palette in palette-based images and hierarchical scrambling. Section 3.2 presents the comparison with other related works. Section 3.3 provides the proposed algorithm including key assignment and key delivery. Section 3.4 gives the experimental results by analyzing the image quality and the analysis of different image quality metrics. Finally, this chapter is concluded in Section 3.5 by highlighting the actual contribution of the proposed algorithm.

Chapter 4 introduces an integrated model of two image protection techniques [22, 58]. Section 4.1 describes BPBE scheme and RDH briefly. Section 4.2 in-

troduces the proposed encryption and embedding process, key derivation, and decryption and extraction process, respectively. Section 4.3 gives the experimental results and analysis by discussing key space, resilience against Jigsaw Puzzle Solvers (JPSs), and compression efficiency. Section 4.4 summarizes the chapter by providing the advantages of the proposed scheme.

Chapter 5 concludes with the major findings of the entire work and suggests the directions for future studies in this area.

Chapter 2

Technical Requirements

In this chapter, the author describes the related technical terms of this dissertation. A hierarchical access control that allows different access rights for different users at various permission levels is more elaborately discussed. The digital image formats are also discussed briefly to be familiar with different image formats. The author uses Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) to measure the quality of distorted images. Therefore, the formulas to calculate MSE, PSNR, and SSIM are explained in this chapter. The process of RDH and BPBE are also discussed because these methods are utilized in this dissertation. The properties of cryptographic hash functions are explained below because they are utilized in key derivation in this dissertation. Finally, the definition of both lossy and lossless compression methods are discussed as below.

2.1 Hierarchical access control system

The hierarchical access control system is a model, where there is an organization of various users in a hierarchy. Generally, the hierarchical structures are important because some of the users have higher access rights than others in different organizational structures. These models are widely used in various application

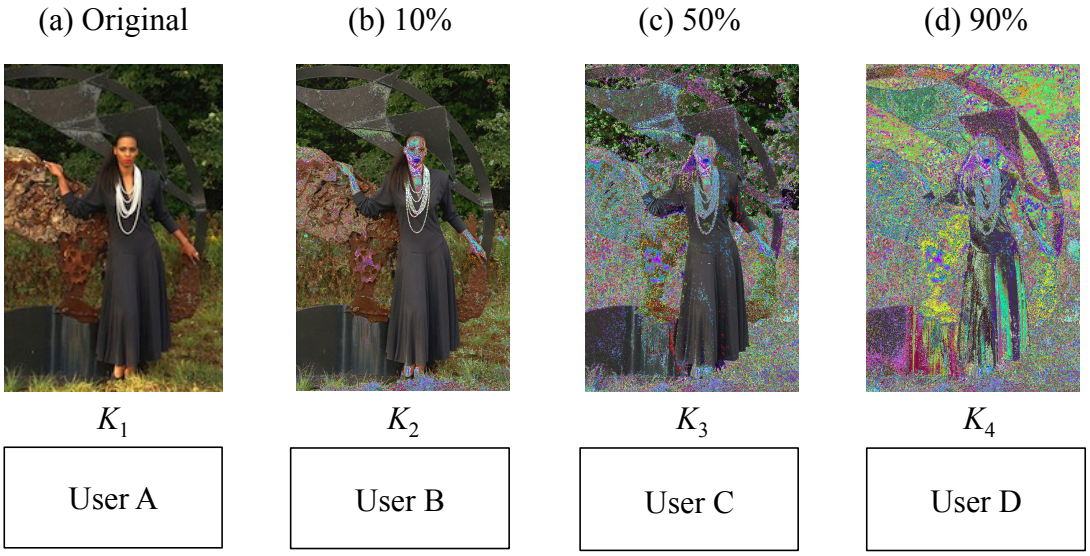


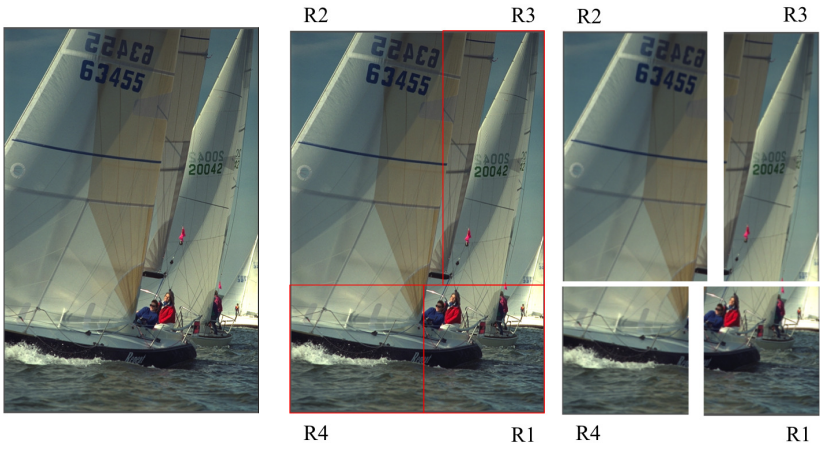
Figure 2.1: Hierarchical encryption.

areas, such as militaries, database management systems, computer networks, government communications, medical systems, and so on. Each user of a hierarchical model is allowed with their own access rights. On another front, the codestream of JPEG2000 [59], which is an international standard for image compression, has a hierarchical structure according to the layers, the resolution levels, the tiles, and so forth. It can be used in different areas, such as digital cinema, broadcasting, image archives and so on. In consequence, the main advantage of this model is to control the access rights according to different permission levels.

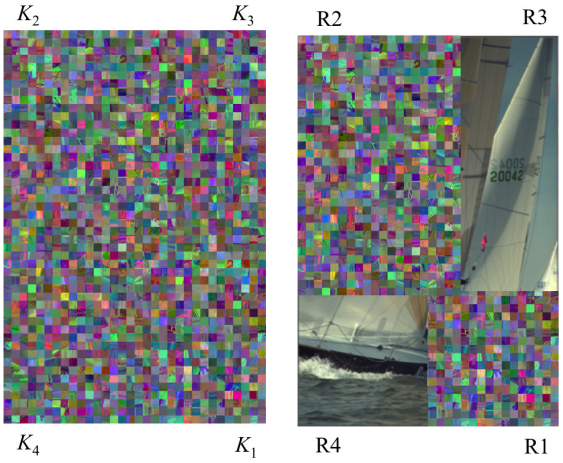
The hierarchical encryption of a digital image is the process of repetitive encryption to the original image in a hierarchical manner. As shown in Fig. 2.1, the original image is hierarchically encrypted to the images with various qualities. The hierarchical key assignment is a method that assigns the encryption keys to each process in a hierarchical series. Thus, each user will use the proper encryption key to decrypt the corresponding images. An original image is hierarchically encrypted to 10%, 50%, and 90% scrambled images as given in Fig. 2.1. The order of access right decreases hierarchically from a user A to a user D. The user A is only the authorized person to get full access right. Therefore, he is able to

access all information with proper key K_1. The user B would obtain 10% scrambled image, and is assumed to access all information by using key K_2 except some information. On the other hand, the user C obtains 50% scrambled image, and access limited information by using key K_3. Finally, the user D would get 90% scrambled image, and is only allowed to access the fewest information using key K_4. Therefore, the images with various qualities can be distributed hierarchically to achieve appropriate access control.

The hierarchical encryption can also be performed on selected regions of an original image [60, 61] as depicted in Fig. 2.2. An original image consists of confidential information, such as faces of people and flag numbers of the boats as given in Fig. 2.2(a). The original image can be divided into four regions, namely, region R1, R2, R3, and R4 based on the confidentiality of information as shown in Fig. 2.2(b). The access rights can be controlled hierarchically according to the priority that is assigned to each regions. We assume R1 is the most confidential region because it has the facial information of people. The regions R2 and R3 that have the flag numbers are considered as the second and third important regions. On the other hand, R4 is the least important region of the original image. Each region is encrypted with their corresponding keys K_1, K_2, K_3, and K_4, respectively. Figure 2.2(c) represents the full encrypted image. For example, a first user is given with the highest priority to get full access rights. Hence, he will obtain all the keys, and retrieve the original image from the final encrypted image. A second user is assumed to access all regions except R1 by using proper key K_2. If a third user would obtain key K_3, then he is able to access R3 and R4 as shown in Fig. 2.2(d). Finally, a fourth user is only permitted to access R4 with proper key K_4 because he has the lowest access right. Therefore, the access rights can be controlled hierarchically in region-based encryption.



(a) Original image (b) Original image divided into four regions



(c) Encrypted image (d) Partially decrypted image

Figure 2.2: Hierarchical encryption for selected regions.

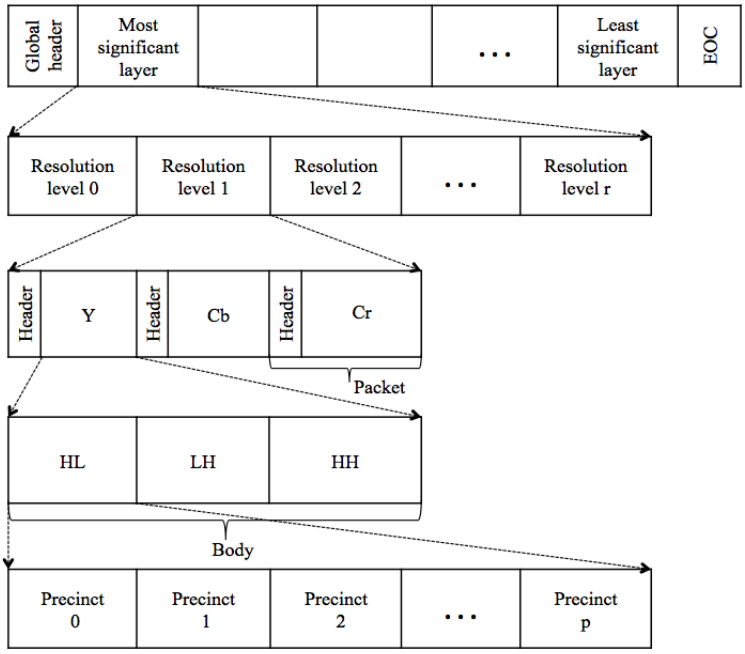


Figure 2.3: Structure of JPEG2000 codestream.

Figure 2.3 shows a structure of JPEG2000 codestream. Each scalability function has its own progression order. The default order is LRCP (Layer-Resolution-Component-Position) as shown in Fig. 2.3. Here, the layer scalability function is the first priority. This means that the first dimension of hierarchy is quality. The global header is followed by sequence of the most significant layer to the least significant layer. Then, the stream is terminated by End of Codestream (EOC). Furthermore, each layer is composed of data for each resolution level. If the original image has color components, then each resolution level is composed of Y, Cb, and Cr components. Resolution level 0 contains only the data of LL, whereas the other resolution levels contain three sub-bands, namely, HL, LH, and HH. These sub-bands contain precincts, which have non-hierarchical positional information. Each packet has a header and a body. It contains partial data for each sub-band. As mentioned above, the codestream of JPEG2000 has a hierarchical structure. It produces different quality images ranging from the lowest quality to the highest quality.

In this dissertation, the author focuses on the access control for image quality as described in the former paragraph.

2.2 Digital image formats

Generally, there are three kinds of image files, i.e., vector, bitmap, and metafiles. The most common image file formats are described as follows.

2.2.1 Bitmap (BMP)

BMP is the native file format of Windows platform. BMP is characterized by the two parameters, namely, the number of pixels, and the color depth. BMP does not allow image compression unless they are converted to any of the JPEG, GIF or PNG. The bitmap image has basically 24-bit color depth.

2.2.2 Graphics Interchange Format (GIF)

GIF is one of the most popular formats for exchanging the graphics. GIF uses only 256 colors, i.e., 8-bit color palette that is made for compact images and consumes less bandwidth. GIF is used in the field of animation since it supports transparency and interlacing.

2.2.3 Portable Network Graphic (PNG)

PNG is a bitmap image, which allows lossless data compression. It is an alternative to GIF and handles transparency more effective than GIF. It offers more color range than GIF by extending up to 24-bit color depth. It should be used for a single image, and does not support animations.

2.2.4 Tag Interchange File Format (TIFF)

TIFF is mainly tag-based international standard that is used for storing and interchanging bitmaps between the hardware and the applications. It does not include compression.

2.2.5 Encapsulated PostScript (EPS)

EPS is a metafile format that can be used for bitmap or vector images. When an EPS image is placed in a document, it is possible to scale up or down without any information loss.

2.2.6 Joint Photographic Experts Group (JPEG)

JPEG is the most popular image format. It supports 24-bit color information. JPEG is usually a lossy compression method, and can be saved in different lossy compression levels.

2.3 Qualitative metrics

2.3.1 Mean Square Error

One of the most popular measures of distortion is MSE. It is calculated by averaging the squared intensity of an original image and distorted image pixels as defined below.

$$\text{MSE} = \frac{\sum_{j=1}^{N}\left(\sum_{i=1}^{M}(X_{i,j} - Y_{i,j})^2\right)}{MN}, \tag{2.1}$$

where $(X_{i,j} - Y_{i,j})$ is the difference between the original and the distorted images.

2.3.2 Peak Signal-to-Noise Ratio

PSNR is the ratio between the maximum possible power of a signal to the power of corrupting noise that affects the quality of its representation. This ratio is used for the quality measurement between original and distorted images. The expression to calculate PSNR can be given as below.

$$\text{PSNR} = 10 \log \frac{255^2}{\text{MSE}}. \tag{2.2}$$

2.3.3 Structural Similarity Index

SSIM [62] index is a novel method for measuring the similarity between the original and the distorted images. First of all, the two images are divided into blocks of 8×8 size and these blocks are then converted into vectors. The similarity measurement depends on luminance comparison $l(x, y)$, contrast comparison $c(x, y)$, and structure comparison $s(x, y)$. Suppose x and y are two non-negative image signals, assuming one of the signal to be of the perfect quality. Luminance comparison function $l(x, y)$ is a function of mean intensity as given below.

$$\mu_x = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{2.3}$$

Secondly, for the signal contrast the standard deviation is estimated as follows.

$$\sigma_x = \left(\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu_x)^2 \right)^{\frac{1}{2}}. \tag{2.4}$$

The contrast comparison $c(x, y)$ is the comparison of σ_x and σ_y. The structure comparison $s(x, y)$ is conducted on these normalized signals $\frac{x-\mu_x}{\sigma_x}$ and $\frac{y-\mu_y}{\sigma_y}$. For the similarity measure, the three functions $l(x, y)$, $c(x, y)$ and $s(x, y)$ can be calculated as follows.

For luminance comparison,

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{{\mu^2}_x + {\mu^2}_y + C_1}, \tag{2.5}$$

where C_1 is a constant.

For contrast comparison,

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{{\sigma^2}_x + {\sigma^2}_y + C_2}, \tag{2.6}$$

where C_2 is a constant.

For structure comparison,

$$s(x, y) = \frac{2\sigma_{xy} + C_3}{2\sigma_x\sigma_y + C_3}, \tag{2.7}$$

where C_3 is a constant.

Therefore, the resulting SSIM index can be computed as shown below.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu^2{}_x + \mu^2{}_y + C_1)(\sigma^2{}_x + \sigma^2{}_y + C_2)}. \tag{2.8}$$

Furtheremore, the Mean SSIM (MSSIM) index to evaluate the overall image quality can be calculated as follows.

$$MSSIM(X,Y) = \frac{1}{M}\sum_{j=1}^{M} SSIM(x_j, y_j), \tag{2.9}$$

where X and Y are the reference and the distorted images. x_j and y_j are the image contents at the j-th local window. M is the number of local window of the image.

2.4 Reversible Data Hiding using Histogram Shifting

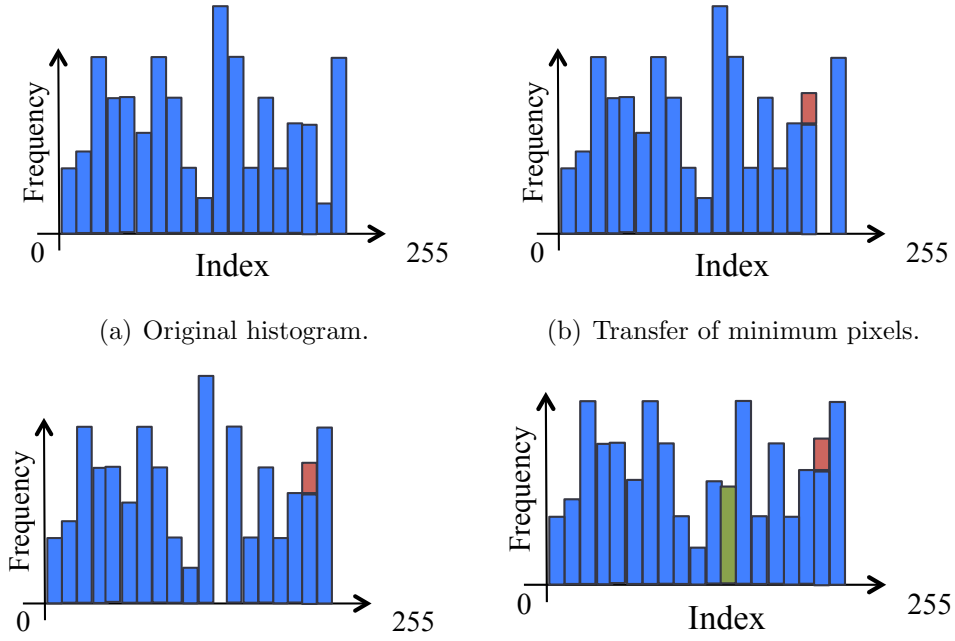
Generally, RDH are the data hiding methods, where the host image is completely recovered after extracting the embedded data. Therefore, the author has applied the conventional RDH algorithm based on Histogram Shifting (HS) [36] to my work as shown in Fig. 2.4. The detail of this method is be described as follows.

1) Generate the histogram $h(x)$ for an $M \times N$ size image.

2) Calculate the minimum point $h(min)$, where $min\epsilon[0, 255]$, and the maximum point $h(max)$, where $max\epsilon[0, 255]$.

3-1) In the case of $max > min$, transfer the minimum bin to the succeeding bin if the minimum point $h(min) > 0$, i.e., the minimum bin pixel values are added to 1.

3-2) In the case of $max < min$, transfer the minimum bin to the preceding bin if the minimum point $h(min) > 0$, i.e., the minimum bin pixel values are subtracted by 1.



(a) Original histogram.

(b) Transfer of minimum pixels.

(c) Shifting the histogram bins.

(d) Changed histogram.

Figure 2.4: Original and changed histograms.

4) Record the number of transferred pixels, and the coordinates (l, k) of those pixels as the side information, which is utilized for the extraction process.

5−1) If $max > min$, decrease the pixel values between $max - 1$ and $min + 1$ by 1.

5−2) If $max < min$, increase the pixel values between $max + 1$ and $min - 1$ by 1.

6−1) In the case of $max > min$, if the to-be-embedded bit is "1", then the pixel value max of the target pixel is changed to $max - 1$. If the to-be-embedded bit is "0", the pixel value remains unchanged.

6−2) In the case of $max < min$, if the to-be-embedded bit is "1", then the pixel value max of the target pixel is changed to $max + 1$. If the to-be-embedded bit is "0" , the pixel value remains unchanged.

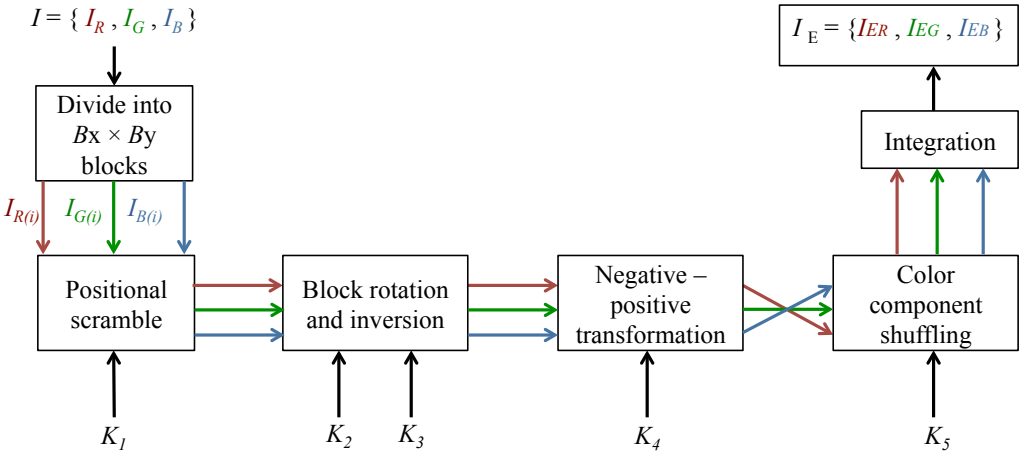


Figure 2.5: Block-permutation-based encryption.

For the color images, the HS method is performed for all R, G, and B components.

2.5 Block-Permutation-Based Encryption Scheme

In BPBE scheme, the original image with $M \times N$ pixels is divided into different non-overlapping blocks of $B_x \times B_y$ pixels. As illustrated in Fig. 2.5, this method has four processes, i.e., positional scramble, block rotation/inversion, negative-positive transformation, and color component shuffling, respectively, which are described as below.

2.5.1 Positional scramble

The positional scramble is the operation that occurs due to the random permutation of the divided blocks by using a random number generated by a key. Figure 2.6 represents the original and the encrypted images by BPBE scheme. Figure 2.6(a) is the original image. Figure 2.6(b) illustrates an example of the positional scramble of the original image when the divided block size is 16×16. The encrypted image will be more difficult to visualize if the block size gets smaller.

2.5.2 Block rotation and inversion

Block rotation is the process when each divided block is rotated to either 0°, 90°, 180°, or 270° randomly. Block inversion is the process that flips each block in four patterns i.e., non-inversion, horizontal inversion, vertical inversion, and both horizontal and vertical inversion. The result of block rotation and inversion is shown in Fig. 2.6(c).

2.5.3 Negative-positive transformation

According to the random number generated by a key, the negative-positive transformation reverses the pixel values in a block. The transformed value p' in the j-th block can be calculated by

$$p' = \begin{cases} p & (r(j) = 0) \\ 255 - p & (r(j) = 1). \end{cases} \tag{2.10}$$

Here, p is the original pixel value and $r(j)$ is a random integer that is given for the j-th block. Figure 2.6(d) shows an encrypted image by using negative-positive transformation.

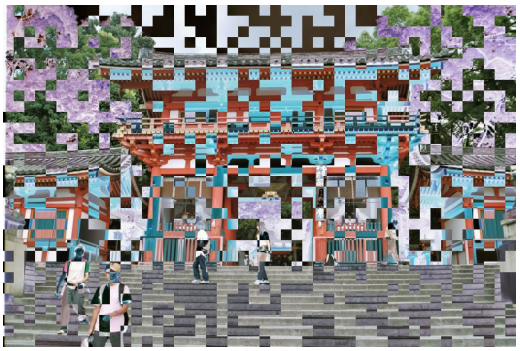
2.5.4 Color component shuffling

As shown in Table 2.1, the color component shuffling is the operation that permutes the values among R, G, and B components according to a random number generated by a key. Figure 2.6(e) shows the result of color component shuffling.

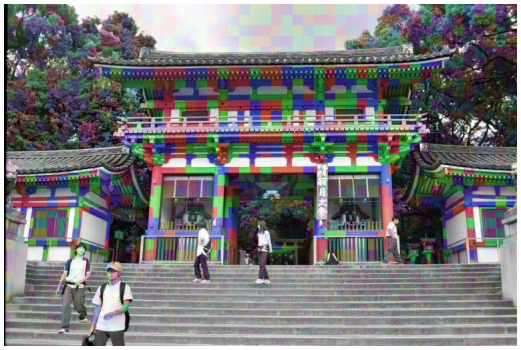
(a) Original image

(b) Positional scramble

(c) Block rotation and inversion



(d) Negative-positive transformation



(e) Color component shuffling

Figure 2.6: Encryption by BPBE scheme.

Table 2.1: Color component shuffling.

Random number	R	G	B
0	R	B	G
1	B	R	G
2	B	G	R
3	G	R	B
4	G	B	R
5	R	G	B

2.6 Cryptographic hash functions

The cryptographic hash functions [63] are one of the most important tools that are used for the security purposes such as authenticity, steganography, and digital signatures. A cryptographic hash function H is a map from the variable-length input bit strings to the fixed-length output bit strings, i.e., $H : (0,1)^* \rightarrow (0,1)^n$. Furthermore, it can be broadly classified into keyed hash functions that uses a secret key, and unkeyed hash functions that does not uses a secret key as shown in Fig. 2.7. The unkeyed hash functions, which are also known as manipulation detection code, can further be classified as one way hash function, collision resistant hash function, and universal one way hash function. In general, the hash functions must satisfy three fundamental security properties as described below.

2.6.1 Collision resistance

It is computationally infeasible to find any pair of two distinct inputs with the same hash value, i.e., $m \neq m^*$, but $H(m)=H(m^*)$.

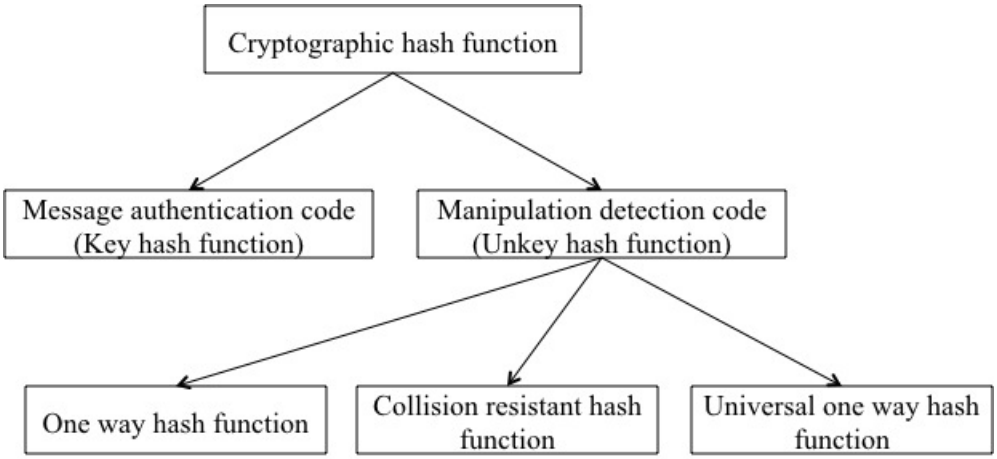


Figure 2.7: Structure of cryptographic hash function.

2.6.2 Preimage resistance

Given a hash value h, it should be computationally infeasible to find an input string m such that $H(m)=h$.

2.6.3 Second-preimage resistance

Given an input string m, it is computationally infeasible to find another input string $n \neq m$ such that $H(n)=H(m)$.

2.7 Compression techniques

Generally, the compression techniques provides the benefits of storage space. The compression procedures can be classified into two categories as defined below.

2.7.1 Lossless compression

The lossless compression minimizes the number of bits that is required to represent the input signal without any loss of information. The application areas are related to medical images, database records, military applications etc. JPEG-LS [64] is a lossless compression method.

2.7.2 Lossy compression

The lossy compression minimizes the number of bits that is required to represent the input signal with some loss of information. However, the compression performance is higher than the lossless compression. This method is used by applications, where they do not require the reconstruction of original data such as CCTV, DVDs, satellite TV etc. JPEG is a lossy compression method.

Chapter 3

Hierarchical Scrambling Method for Palette-Based Images Using Bitwise Operation

The current growth of the Internet has led to sharing a huge amount of digital media and content among various communication channels. Hence, there is always a concern about information leakage and privacy. The digital contents can be securely transmitted by using various methods. Generally, there are different cryptographic techniques to encrypt the digital content and make it extremely difficult to decode by unauthorized users. There are various types of scrambling methods that are used for image protection. Digital watermarking can be used to hide copyright information into digital media such as images, audios, and videos. However, it has limitations on unauthorized duplications because it does not degrade the image quality visually. On the other hand, image scrambling makes an original image into a visually degraded image. Full encryption such as AES and triple DES are the robust ways to encrypt the images. However, this way of encryption does not allow the users to have a partial viewing and the proper evaluation without its proper keys. It is also not possible to decrypt the original content correctly if there exist any errors. Therefore, the full encryption may not be always the best way to protect the copyright and the privacy. The perceptual encryption, which is one of the secure encryption methods, scrambles only a part of the image. Any users are allowed to view the degraded image and analyze it.

In contrast, the authorized users can only decrypt the image by using its proper key.

As mentioned in Chapter 1, the palette-based images are limited to 256 colors. Therefore, the amount of data is less than that of 24-bit colors. These images are used in different application areas such as websites and multimedia. Furthermore, electronic paper (e-paper) has been very popular lately. However, the power consumption of full color is larger as more colors need to be displayed. Thus, the palette-based images are important because they require less power consumption to display the limited colors. Considering the importance of encryption in palette-based images, a flexible partial encryption for them has been proposed [10]. However, the pixel scrambling method is not based on hierarchical access control in this method. Similarly, the conventional method [19] uses cyclic shift operations that generate only a few patterns, and the method is not secure. A partial scrambling method [20] was proposed, which generates fewer patterns and thus is quite predictable. Hence, these methods do not provide enough security to prevent malicious attacks by unauthorized users.

In this chapter, the author proposes a hierarchical scrambling method for palette-based images using bitwise operation [21]. The number of target color components (Red (R), Green (G), and Blue (B)) and the target pixel threshold are the major parameters for the quality control of scrambled images. The images in the proposed method are more distorted than the conventional methods [19, 20]. Furthermore, the author introduces a hierarchical key assignment scheme to provide a secure environment against unauthorized decryption.

3.1 Technical preliminaries

The related technical terms are described below before discussing the proposed method. The colors in the palette are denoted as entities. Each entity is assigned to an index number. In an image, each pixel has an index number that indicates the corresponding entity, which specifies the RGB color.

3.1.1 Composition of color palette in palette-based images

Each index number i of a color palette contains information specifying the location of the entity in the palette as shown in Fig. 3.1. The color palette is an array consisting of three primary colors R, G, and B, and N rows. Here, N is the number of entities in the palette. The image is an 8-bit image when $N = 256$.

3.1.2 Hierarchical scrambling

Let us assume that the scalable medium has two-dimensional scalability ($D = 2$). One of the dimensions is the ratio of the pixels to be scrambled. Another dimension is the R, G, and B color components. The hierarchical scrambling of two-dimensional scalability is shown in Fig. 3.2. $R_{100}G_{100}B_{100}$ is the expected lowest quality image. The process of scrambling is performed on each entity with its corresponding key that is generated by a hierarchical key assignment scheme.

3.2 Conventional work

3.2.1 Hierarchical scrambling of palette-based images

This method [19] cyclically shifts the binary digits to scramble the images. The following steps show the process of hierarchical scrambling of palette-based images.

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

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

Step 3 Sort all entities E_i in ascending or descending order based on their frequencies F_i as shown in Fig. 3.3.

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

Palette-based image

Index = 2

Index i	R	G	B	Frequency
0	58	42	40	600
1	104	105	69	10000
2	20	94	42	500
3	113	60	81	7000
4	56	88	10	7235
…	…	…	…	…
255	99	149	108	9878

Color palette

Figure 3.1: Structure of palette-based image.

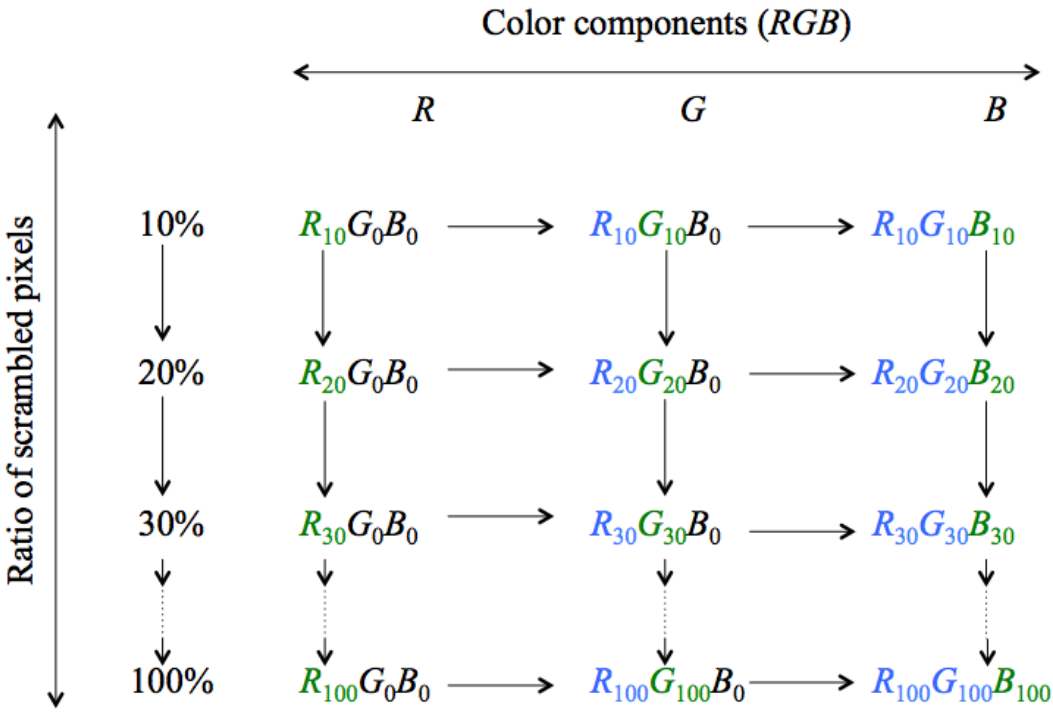


Figure 3.2: Hierarchical scrambling of two-dimensional scalability ($D = 2$).

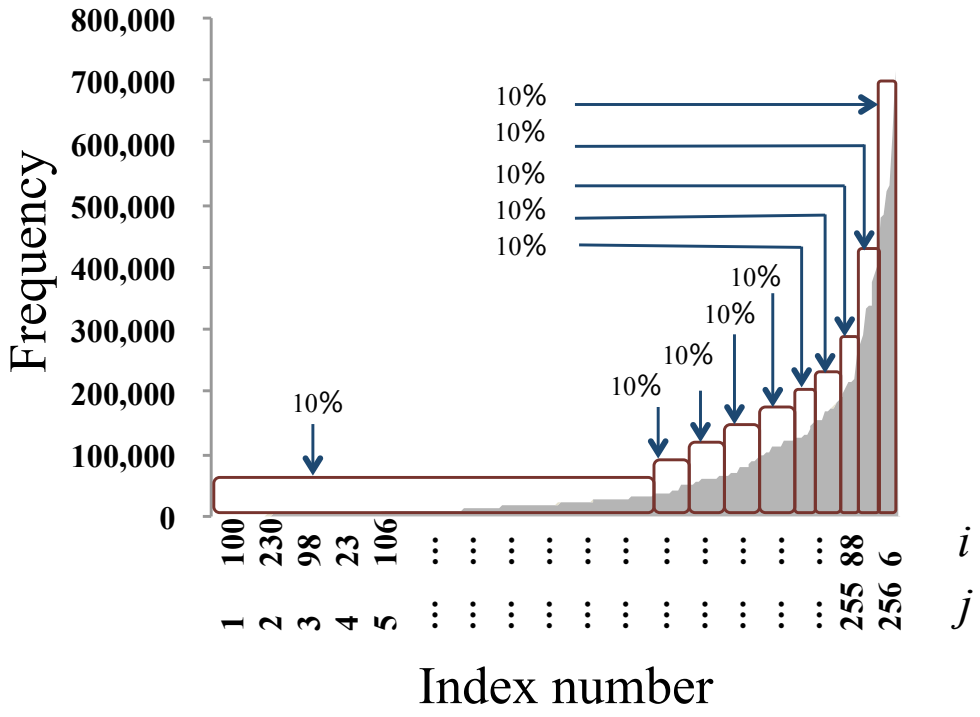


Figure 3.3: Histogram sorted in ascending order.

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 example, the decimal value "94" is chosen to be manipulated and is converted into the binary value equivalent of $(01011110)_2$ as illustrated in Fig. 3.4. 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. If the random number is 1, the rightmost two digits are cyclically shifted to the left. Accordingly, the binary value becomes $(10010111)_2$, which is equivalent to "151" as given in Fig. 3.4. Therefore, this method uses cyclic shifts to operate the binary bits and scramble the images.

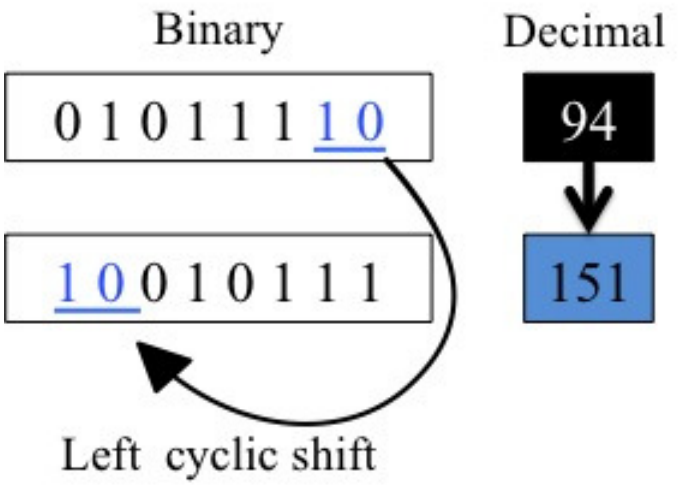


Figure 3.4: Cyclic shift operation.

The main disadvantages of this approach are: 1) due to the use of cyclic shift operation, we can change the current color in the palette to another color only from a limited range of colors and 2) the cyclic shift operation is very simple to perform. Consequently, this method is vulnerable to malicious attacks, such as brute-force attacks [65]. Therefore, the main objective of this dissertation is to propose a more secure method.

3.2.2 Hierarchical scrambling of palette-based images using transposition cipher

The principal idea of the conventional method [20] is based on changing the positions of the R, G, and B values. In this method, Steps 1 to 5 of 3.2.1 are carried out. Then, the positions of the RGB values of the target entities are randomly shuffled to form the scrambled image as depicted in Fig. 3.5. However, we can generate only a few numbers of patterns as this scrambling process is too simple. Hence, it is hard to achieve adequate complexity in the decryption process to thwart unauthorized users.

Figure 3.5: Procedure of conventional method based on transposition cipher.

3.3 Proposed method

A hierarchical scrambling method for palette-based images using bitwise operation is proposed in this section [21]. The proposed method is better than the conventional methods [19, 20] in terms of the security against malicious users as well as the distortion of the scrambled images.

3.3.1 Parameters for quality control

The author introduces two parameters, namely, target pixels and target color components to control the quality of the scrambled images.

3.3.1.1 Target pixels

Figure 3.1 shows an example of the relation between the index number of entities and their use of frequency. Figure 3.3 illustrates a color palette sorted in

ascending order and the process of choosing the frequencies of pixels in a hierarchical manner. The proposed method can control the quality of scrambled images and select scrambling from low-frequency regions if the color palette is sorted in descending order or from high-frequency regions if the color palette is sorted in ascending order.

3.3.1.2 Target color components

Figure 3.2 shows the process of manipulating the target RGB values. For example, 10% of the R-component is scrambled first and is denoted as the R_{10} component. The scrambled image proceeds to the right to scramble 10% of the G-component. The resulting 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. For 20% scrambling, the R_{10} component is passed below for the next 10% scrambling to obtain the R_{20} scrambled image. The resultant R_{20} is passed to the operation on the right, and the G_{10} component is passed below for the next 10% scrambling to obtain R_{20} and G_{20} components. The resultant R_{20} and G_{20} components are passed to the operation on the right, and the B_{10} component is passed below for the next 10% scrambling to obtain the scrambled image with R_{20}, G_{20}, and B_{20} components. This process of scrambling is repeated for all the pixels.

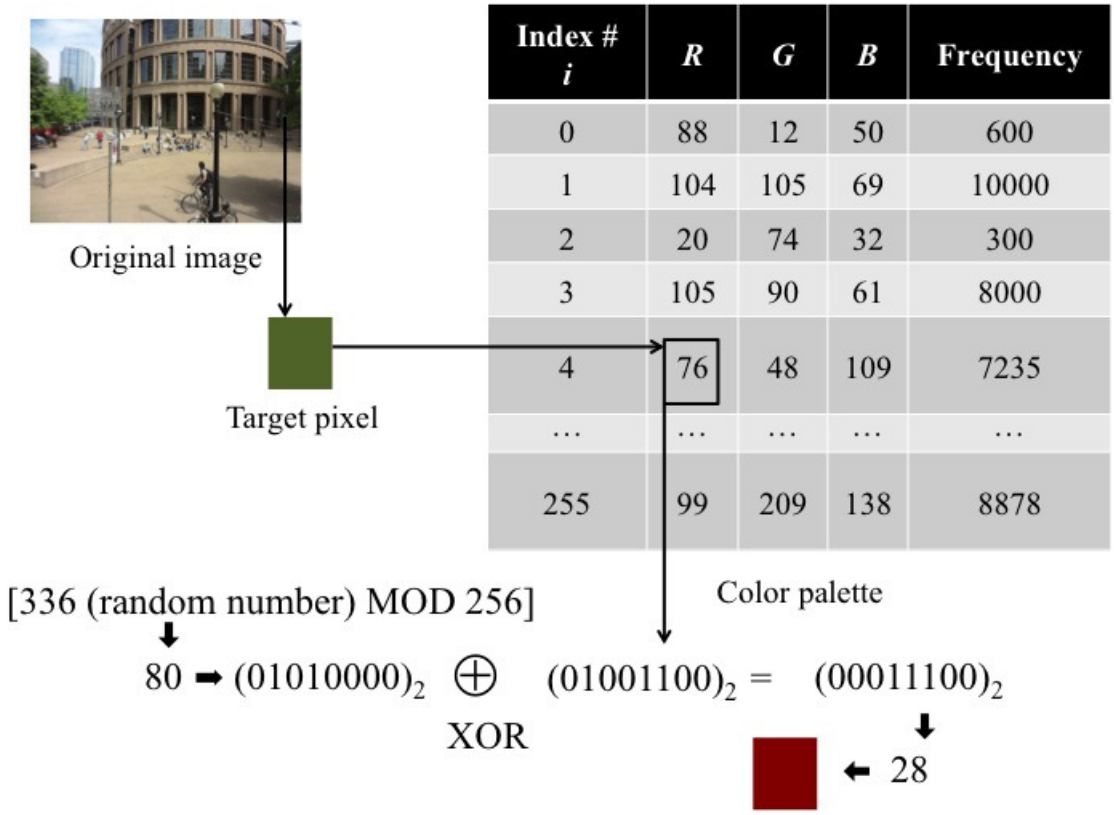


Figure 3.6: Overview of proposed method.

3.3.2 Algorithm

Figure 3.6 shows the structure of the proposed method. The scrambling process is described in detail as follows.

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

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

Step 3 Sort all entities E_i in ascending or descending order based on their frequencies F_i as given in Fig. 3.3.

Step 4 Change 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 a modulo operation with respect to N, where N is 256. The decimal result is then converted into its binary value as shown in Fig. 3.6.

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 components and the target entities components are modified.

For this method, Steps 1 to 6 are carried out as described above in 3.2.1. It is noted that the author adopts XOR operation as the bitwise operation in this dissertation.

3.3.3 Key assignment and encryption

As shown in Fig. 3.7, the hierarchical key assignment schemes using Simple Hash

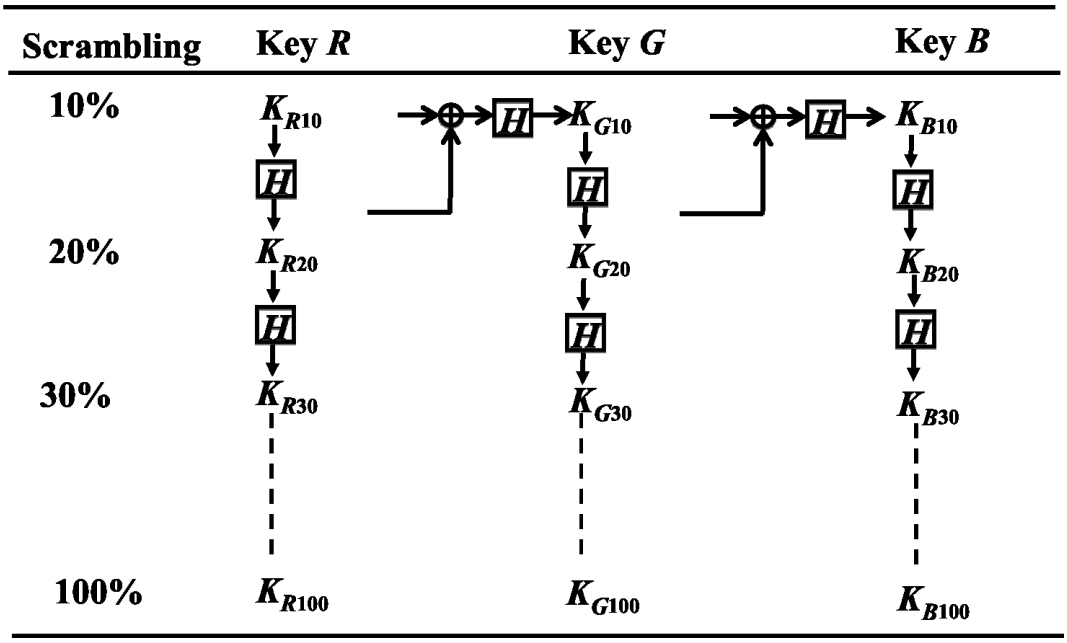


Figure 3.7: Key assignment using SHCs and RHCs.

Chains (SHCs) and Recursive Hash Chains (RHCs) [24] has been adopted. Fundamentally, each scrambling class is assigned by a proper key, which can generate other required keys for all lower classes in the hierarchy.

The codestream is encrypted entity-by-entity by using different keys in access control for scalable media. Figure 3.7 shows the key derivation for encryption. For instance, keys K_{R10}, K_{G10}, and K_{B10} are the keys for the R_{10}, G_{10}, and B_{10} components, respectively. The managed key is K_{R10}. The derivation of K_{R20} from K_{R10} can be given as follows.

$$K_{R20} = H(K_{R10}). \tag{3.1}$$

Here, $H(\cdot)$ is a cryptographic one-way hash function, i.e., SHA-256 [66]. The other keys K_{Rs} (s = 30, 40, ..., 100) are also derived with SHCs. In contrast, keys K_{G10} and K_{B10} are derived by RHCs as given by

$$K_{G10} = H(f(K_{R10}, H(K_{R10}))) \tag{3.2}$$

$$= H(f(K_{R10}, K_{R20})), \tag{3.3}$$

and

$$K_{B10} = H(f(K_{G10}, H(K_{G10}))) \tag{3.4}$$

$$= H(f(K_{G10}, K_{G20})), \tag{3.5}$$

where $f(\cdot)$ is a function, which represents bitwise exclusive OR operation. By using the hash chains, the multiple keys can be recursively computed from the master key. The main advantages of using hash chains are: 1) they can be independent from each other, which make the security robust against collusion attacks and 2) they can diminish the complexity of key management and delivery. In this case, these keys are the seeds of pseudo-random numbers. Additionally, the key management for storing single key K_{R10} is simple. It is noted that the decryption process requires both the the keys and the pseudo-random numbers.

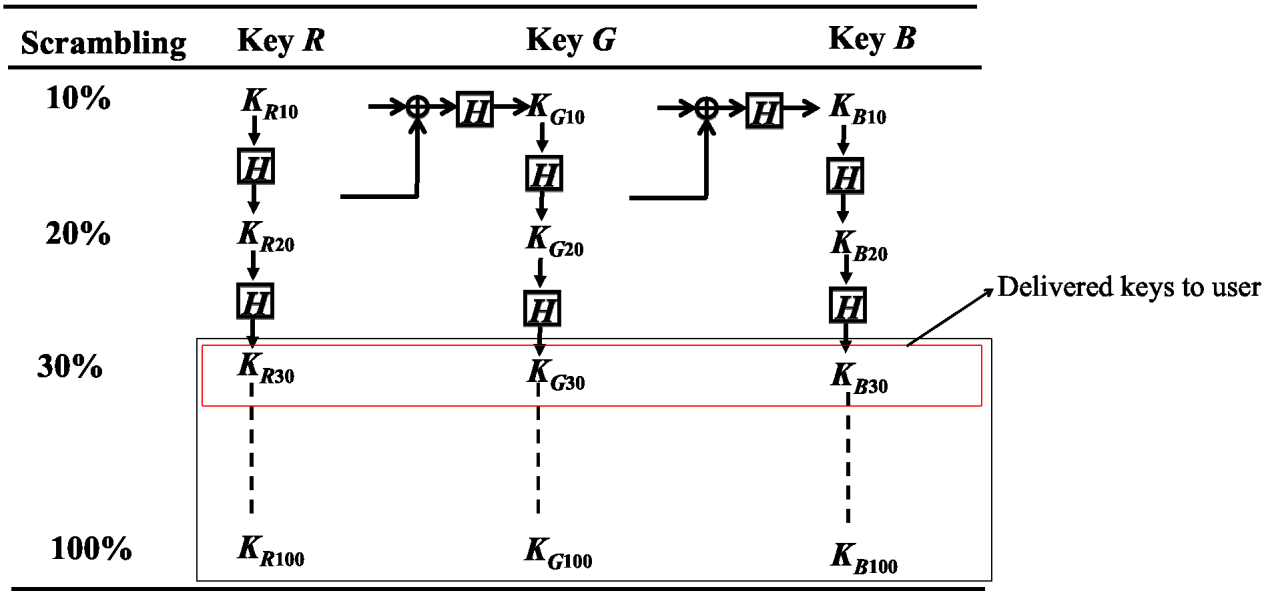


Figure 3.8: Decryption process for image quality of $R_{20}G_{20}B_{20}$.

3.3.4 Key delivery and decryption

As represented in Fig. 3.8, 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 obtained from the three delivered keys by using SHCs. Similarly, the user can decrypt the corresponding 24 components, e.g. R_{30}, G_{30}, ..., B_{100} components.

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} as shown in Fig. 3.9. The 9 keys are derived from the delivered key using SHCs. The corresponding 10 components such as B_{10}, B_{20}, ..., B_{100} components can be decrypted by the user.

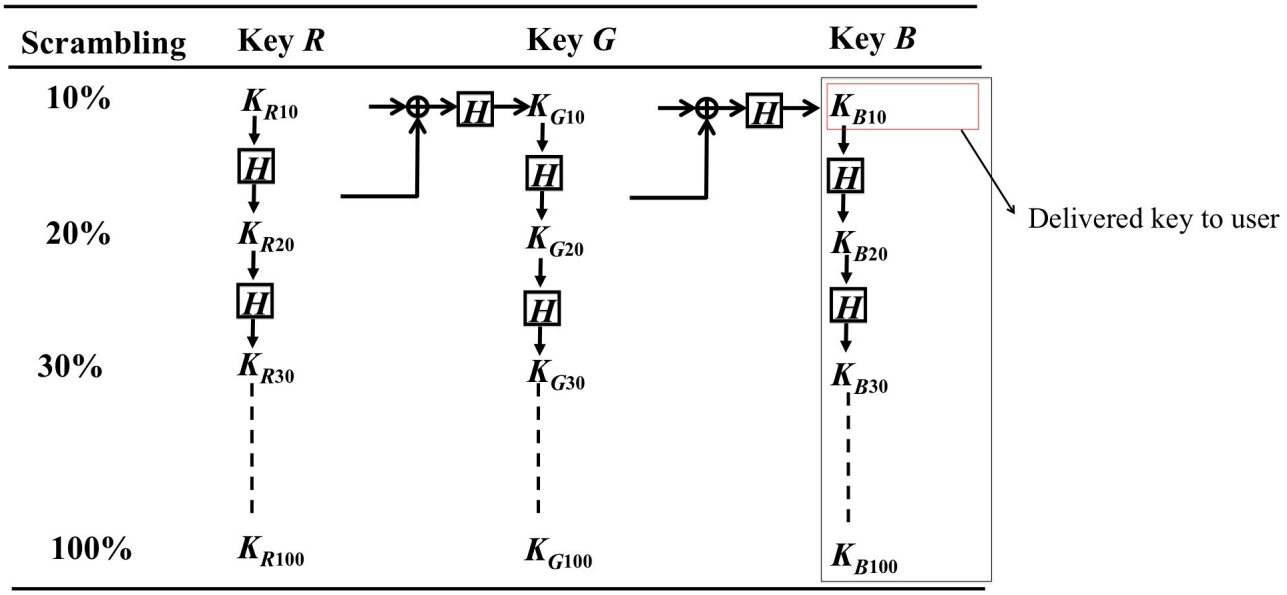


Figure 3.9: Decryption process for image quality of $R_{100}G_{100}B_0$.

3.4 Experimental results

The proposed and conventional methods [19,20] were tested by using ten different images available from the kodak lossless true color image suite [67] as shown in Fig. 3.10. The original images are of two different sizes: 768×512 and 512×768 pixels. While considering the security, the private text in the images, such as the number plate of an automobile, can be one of the important factor. Therefore, the author used the original images such as kodim03, kodim08, kodim09, kodim10, and kodim14 to check whether the text in those images is recognizable after scrambling or not. Furthermore, as depicted in Fig. 3.10, the author also considered images with small (kodim12 and 14), medium (kodim18), and large faces (kodim04 and 15) for the simulation to evaluate the dominant ratio of human faces in the image.

The colors of the 24-bit images were reduced before scrambling to perform the simulation. By calling matlab function "rgb2ind" and specifying the maximum number of colors in the output such as 32 (5-bit), 64 (6-bit), 128 (7-bit), and 256

(a) kodim03

(b) kodim05

(c) kodim08

(d) kodim12

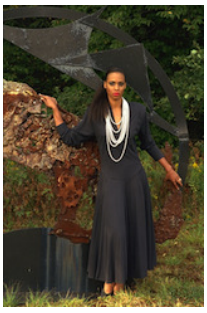
(e) kodim14

(f) kodim15

(g) kodim04

(h) kodim09

(i) kodim10



(j) kodim18

Figure 3.10: Original images from kodak lossless true color image suite.

(8-bit) colors without setting dithering option, the minimum variance quantization was performed on the original images. The author evaluates the performance using PSNR, SSIM, and MSE between the visually distorted images and the original image for the proposed and the conventional methods. The parameters used for calculating the SSIM values were given as $Window$ (local window value) = 11, K (constants in the SSIM index formula) = [0.01, 0.03], and L (dynamic range of image) = 1.

3.4.1 Analysis of image quality

Figure 3.11 represents 30%, 60%, and 100% scrambling of the original image "kodim09" when the maximum number of colors is 256. With the increase in percentage of target pixels, the quality of the scrambled images is gradually degraded. The text included in the original image "kodim09" is also gradually degraded and is difficult to read because the percentage of scrambling gets increased. 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 are depicted in Fig. 3.12. We can observe that Figs. 3.12(a), 3.12(b), and 3.12(c) are different from each other. For instance, the text in the image of Fig. 3.12(a) has been distorted the most as it is difficult to recognize the text. In contrast, Fig. 3.12(c) has been distorted less as it is easily readable. Thus, the quality of the distorted images can be controlled by using different seeds. The monochrome version of the scrambled images of image "kodim09" with the intensity level as 32 is shown in Fig. 3.13. The proposed method is performed for "kodim09" after converting to monochrome version. We can see that the quality of the scrambled image is distorted hierarchically for the monochrome image also. Figure 3.14 represents the full scrambled images of "kodim09" (monochrome) with 32 intensity levels by using three different seeds. It is hard to see whether people are present in the ships or not in those scrambled images. The text written on the flag of the ship that is present at the backside, are completely scrambled and are invisible. On the other hand, the upper text written on the flag of ship that is present at the front is visible, whereas the lower text is distorted considerably, and is complicated to read the numbers.

Figure 3.15 illustrates 30%, 60%, and 100% scrambling of the original image "kodim18" when the maximum number of colors is 256. The image is distorted the most in the case of 100% scrambling. It is noted that the color of the woman′s skin and the background, where she is standing, gradually become distorted with increasing percentage of target pixels. It is hard to see the actual color of her skin and background of image at the quality of $R_{100}G_{100}B_{100}$ scrambling. We can also see that the visual quality is degraded as the number of target color components is increased in the simulation results. In general, the simulation

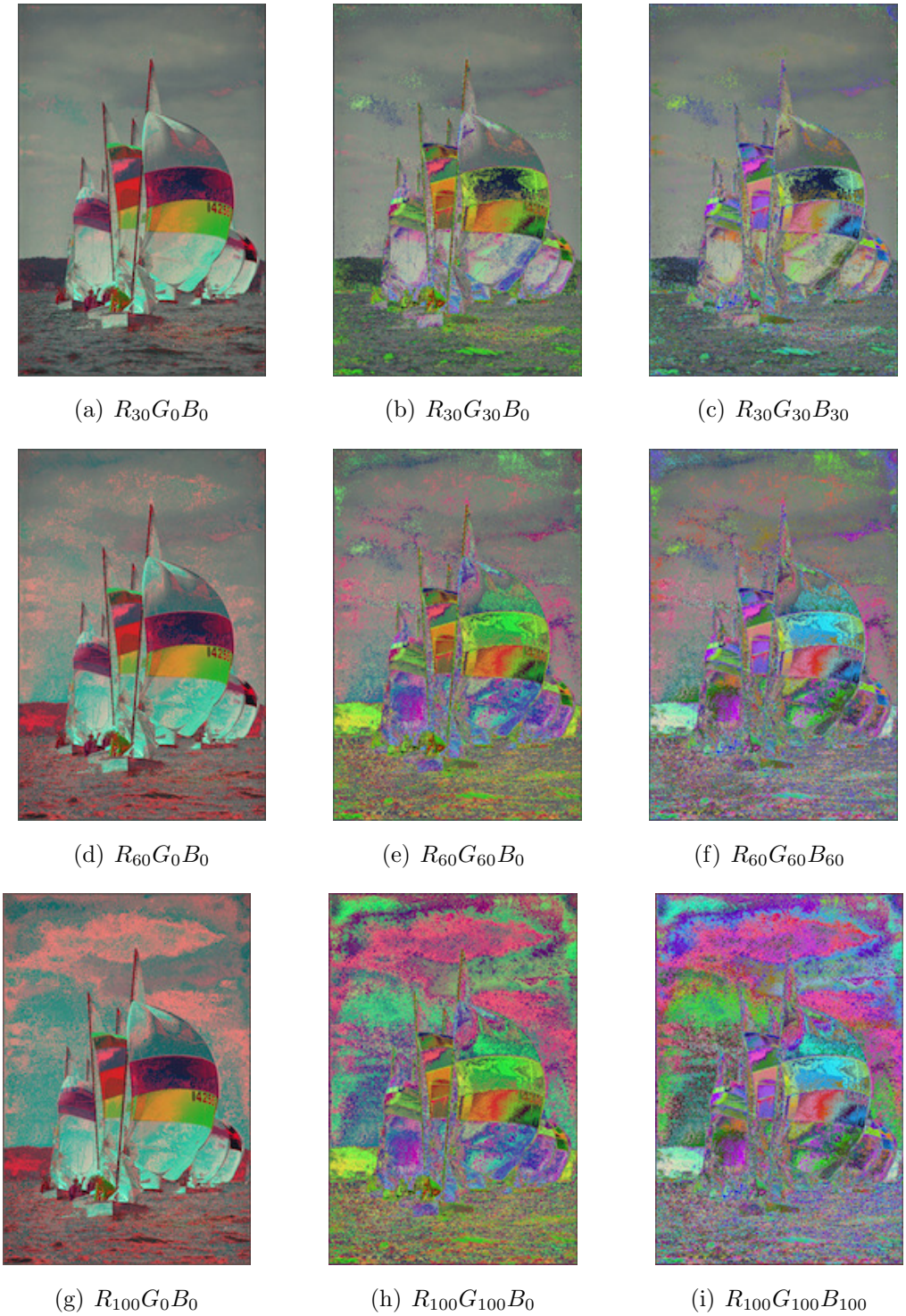
results represent that the hierarchical scrambling scheme is efficient for these test configurations. Alternatively, the image quality is degraded in a hierarchical manner as the percentage of target pixels gets increased. Figure 3.16 depicts the three scrambled images of "kodim18" with 32 colors, where the quality is $R_{100}G_{100}B_{100}$, using different seeds. Figures 3.16(a), 3.16(b), and 3.16(c) are different from each other. It is hard 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. The monochrome version of the scrambled images of original image "kodim18" with 32 intensity levels is depicted in Fig. 3.17. Figure. 3.18 represents the full scrambled images of "kodim18" (monochrome) with 32 intensity levels by using three different seeds. The simulation results show that the face of the woman is distorted and is difficult to see.

3.4.2 Analysis of SSIM, PSNR, and MSE metric

The author measured the quantitative performance of the proposed scheme by using the PSNR, SSIM, and MSE values of all the distorted images. The SSIM values 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, are as illustrated in Fig. 3.19. The author controlled two parameters, namely, the target color components and the target pixels in the experiment. The target pixels were selected from the target color components in a hierarchical manner. The author realizes that there is a hierarchical scrambling of images for all quantized colors. Figures 3.20 and 3.21 show the PSNR and MSE values of "kodim18" obtained by scrambling the R, G, and B components for different numbers of colors. We can see that the value of PSNR decreases as the percentage of scrambling increases. As the percentage of scrambling increases, the value of MSE increases. This is because the higher the value of MSE, the higher the distortion level is. Figures 3.22, 3.23, and 3.24 show the graphs of SSIM, PSNR, and MSE, respectively, of "kodim18" with 256 colors by scrambling (R), (R and G), and (R, G, and B) components. Additionally, these graphs show that the scrambling process occurred in a hierarchical manner.

Tables 3.1, 3.2, and 3.3 show the values of SSIM, PSNR, and MSE for "kodim18" with 256 colors where the target color components are the R, G, and B components. Comparatively, the values of SSIM in the proposed method are lower compared with those of the two conventional methods [19, 20]. The proposed method can scramble the images more than the two conventional methods. In addition, the proposed method has the most distorted image as the value of SSIM for 100% scrambling in the proposed method is much lower, i.e. 0.006, than that of other two conventional methods, which have SSIM values of 0.082 and 0.311. The conventional methods also have larger PSNR values than the proposed method. The values of MSE in the proposed method are higher than those of the conventional methods. Thus, the images are more degraded in the proposed method than in the conventional methods [19, 20].

As we can change the current color in the palette to another one from the 24-bit colors, that is, full colors, the proposed method is quite efficient as compared to the above-mentioned conventional methods [19,20]. The conventional method [19] uses a cyclic shift operation to change the binary bits by choosing a random number in the range 0 to 7. As a result, we can change the current color in the palette to another one only from the limited colors but not from the full colors. On the other hand, the positions of the RGB values of target entities are shuffled randomly in the range 1 to 256 in the conventional method [20]. Therefore, we cannot change the colors to one from the full colors.

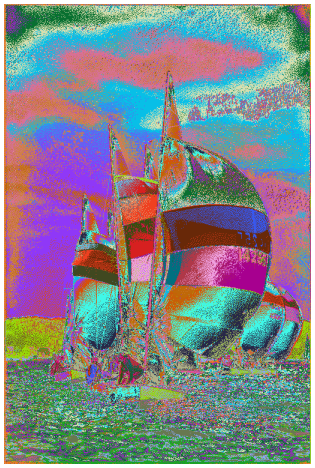


(a) $R_{30}G_0B_0$ (b) $R_{30}G_{30}B_0$ (c) $R_{30}G_{30}B_{30}$

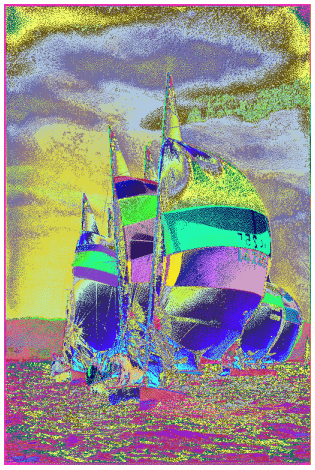
(d) $R_{60}G_0B_0$ (e) $R_{60}G_{60}B_0$ (f) $R_{60}G_{60}B_{60}$

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

Figure 3.11: Scrambled images in proposed method ("kodim09" with 256 colors).



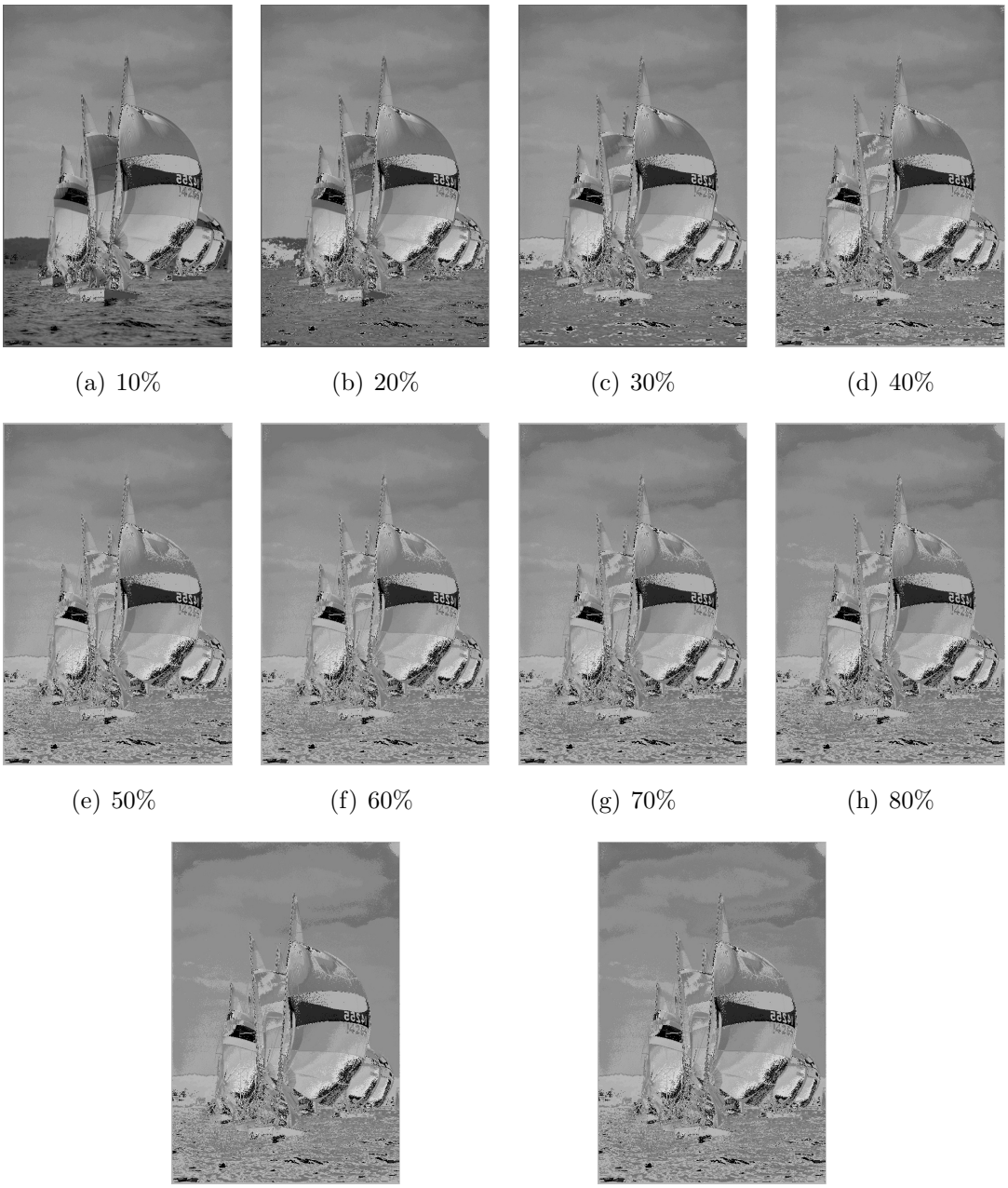
(a) Seeding pattern A



(b) Seeding pattern B

(c) Seeding pattern C

Figure 3.12: Full scrambled images ($R_{100}G_{100}B_{100}$) of "kodim09" with 32 colors in proposed method using three different seeds.



(a) 10% (b) 20% (c) 30% (d) 40%

(e) 50% (f) 60% (g) 70% (h) 80%

(i) 90% (j) 100%

Figure 3.13: Scrambled images in proposed method ("kodim09" with 32 intensity levels, monochrome).

(a) Seeding pattern A

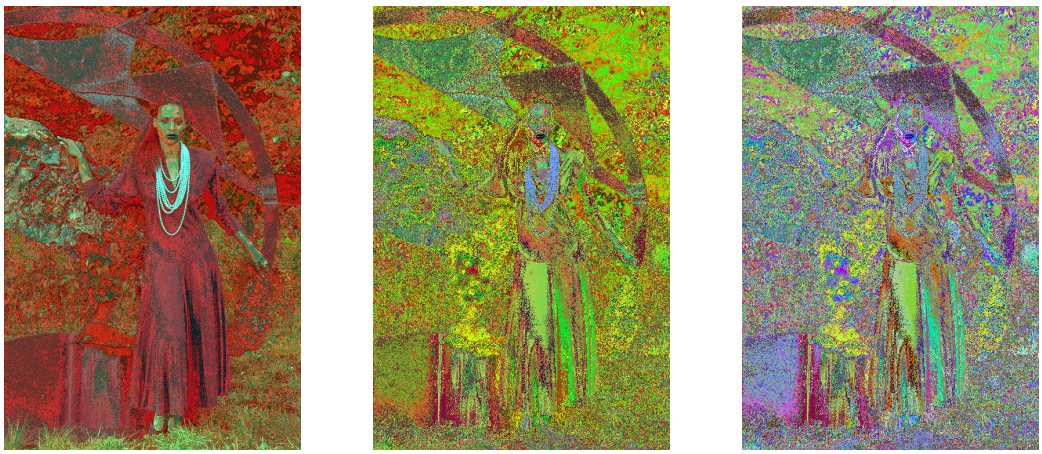
(b) Seeding pattern B

(c) Seeding pattern C

Figure 3.14: Full scrambled images of "kodim09" (monochrome) with 32 intensity levels in proposed method using three different seeds.

(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) $R_{60}G_{60}B_{60}$

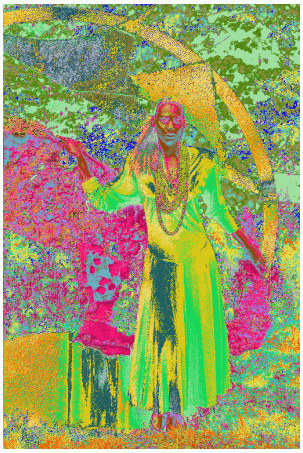


(g) $R_{100}G_0B_0$ (h) $R_{100}G_{100}B_0$ (i) $R_{100}G_{100}B_{100}$

Figure 3.15: Scrambled images in proposed method ("kodim18" with 256 colors).

(a) Seeding pattern A

(b) Seeding pattern B



(c) Seeding pattern C

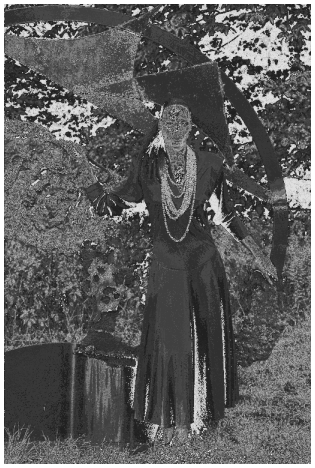
Figure 3.16: Full scrambled images ($R_{100}G_{100}B_{100}$) of "kodim18" with 32 colors in proposed method using three different seeds.

(a) 10% (b) 20% (c) 30% (d) 40%

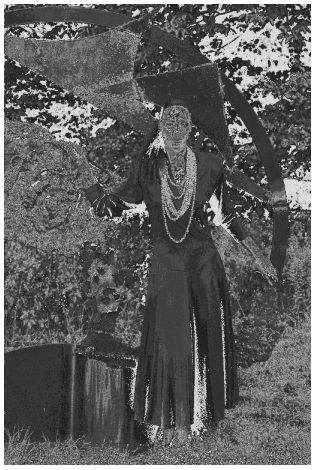
(e) 50% (f) 60% (g) 70% (h) 80%

(i) 90% (j) 100%

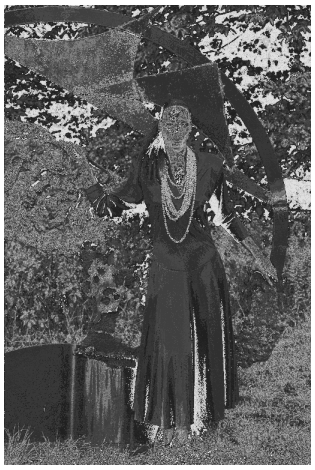
Figure 3.17: Scrambled images in proposed method ("kodim18" with 32 intensity levels, monochrome).



(a) Seeding pattern A



(b) Seeding pattern B



(c) Seeding pattern C

Figure 3.18: Full scrambled images of "kodim18" (monochrome) with 32 intensity levels in proposed method using three different seeds.

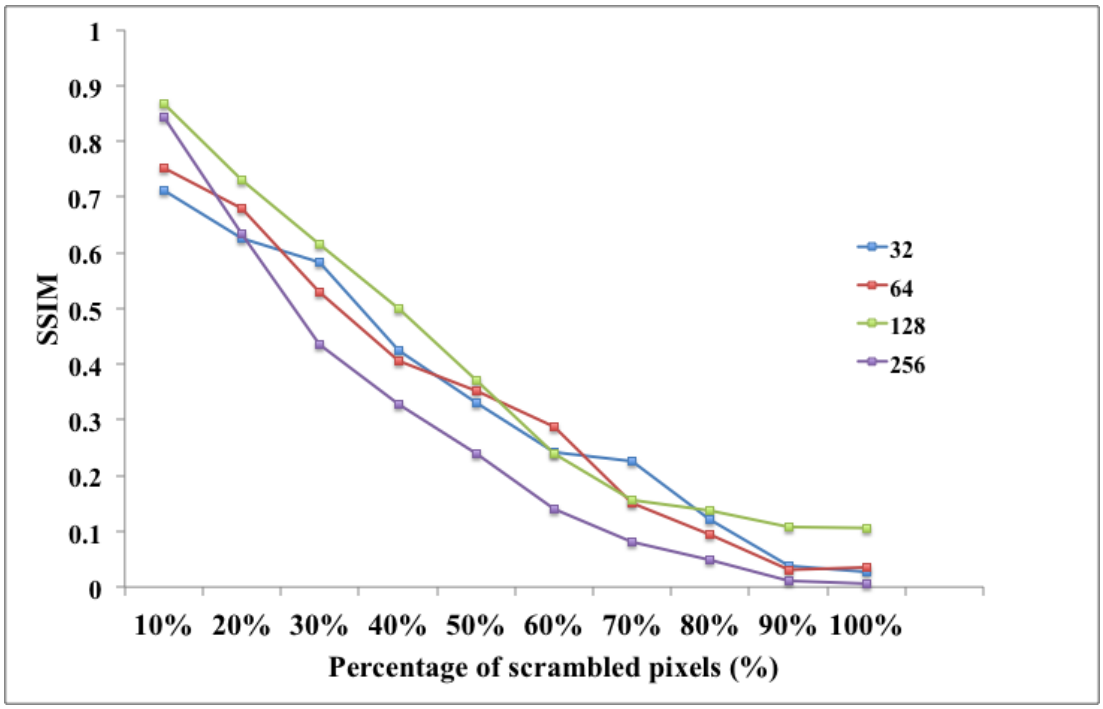


Figure 3.19: SSIM values of "kodim18" obtained by scrambling R, G, and B components for different numbers of colors.

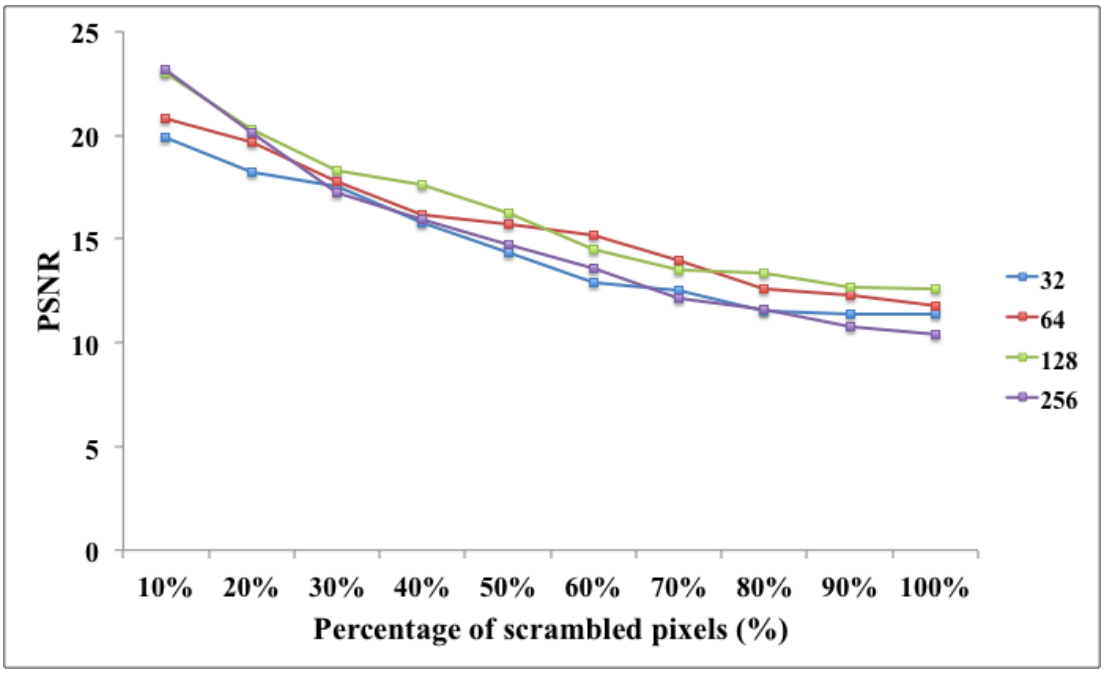


Figure 3.20: PSNR values of "kodim18" obtained by scrambling R, G, and B components for different numbers of colors.

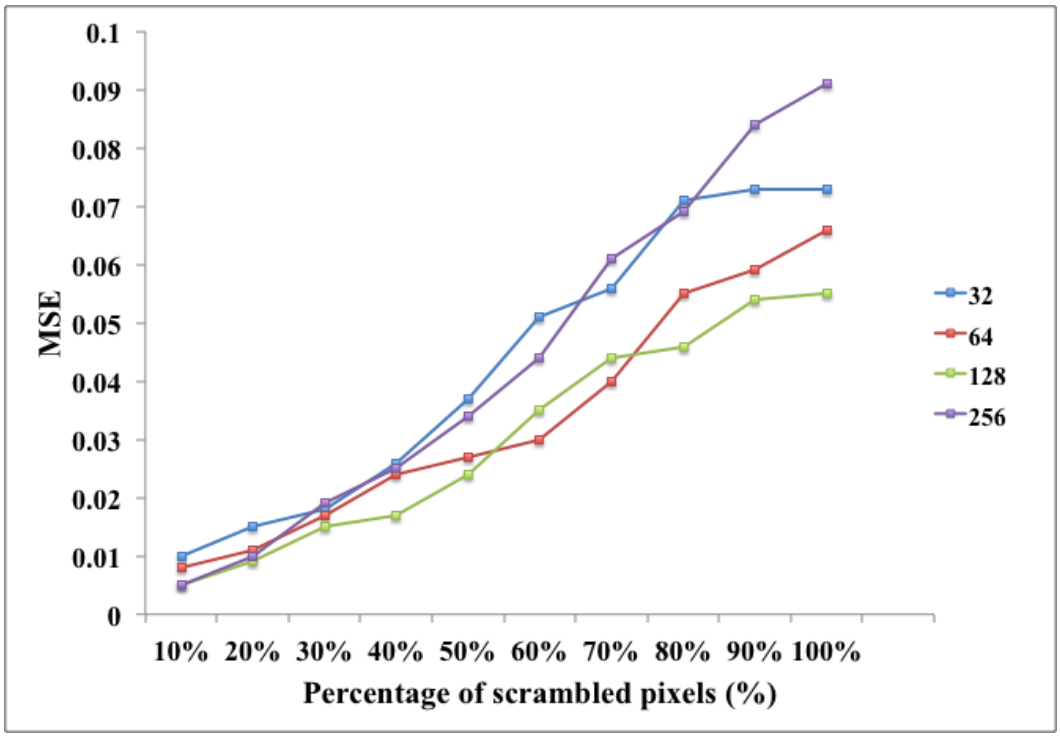


Figure 3.21: MSE values of "kodim18" obtained by scrambling R, G, and B components for different numbers of colors.

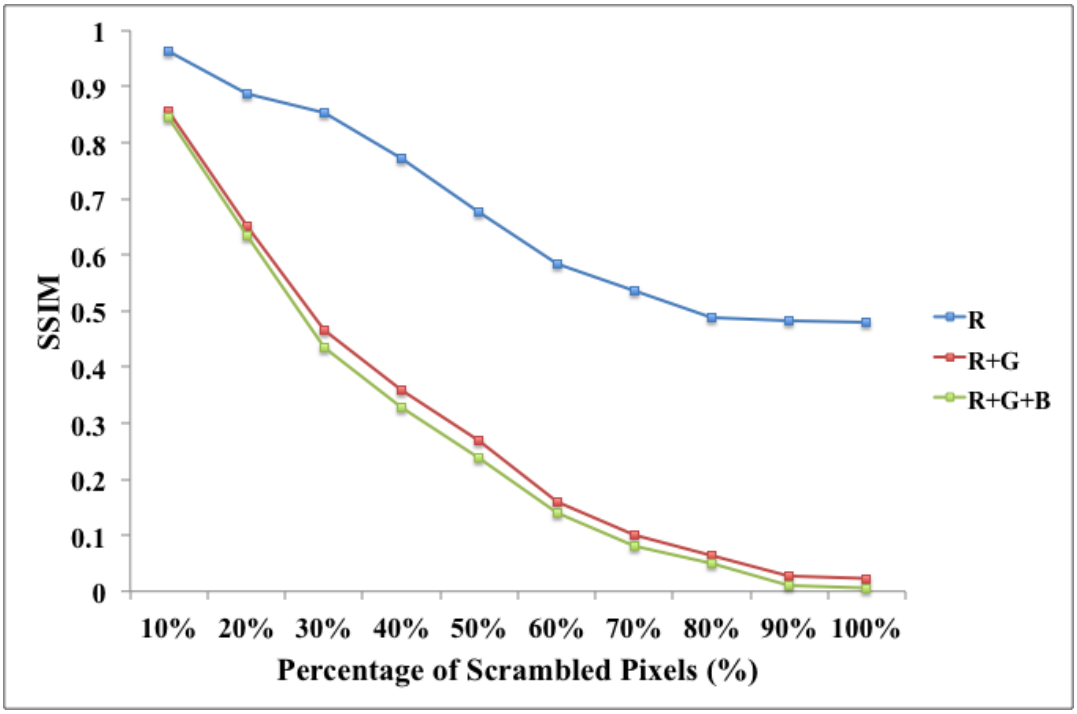


Figure 3.22: SSIM values of "kodim18" with 256 colors obtained by scrambling (R), (R and G), and (R, G, and B) components.

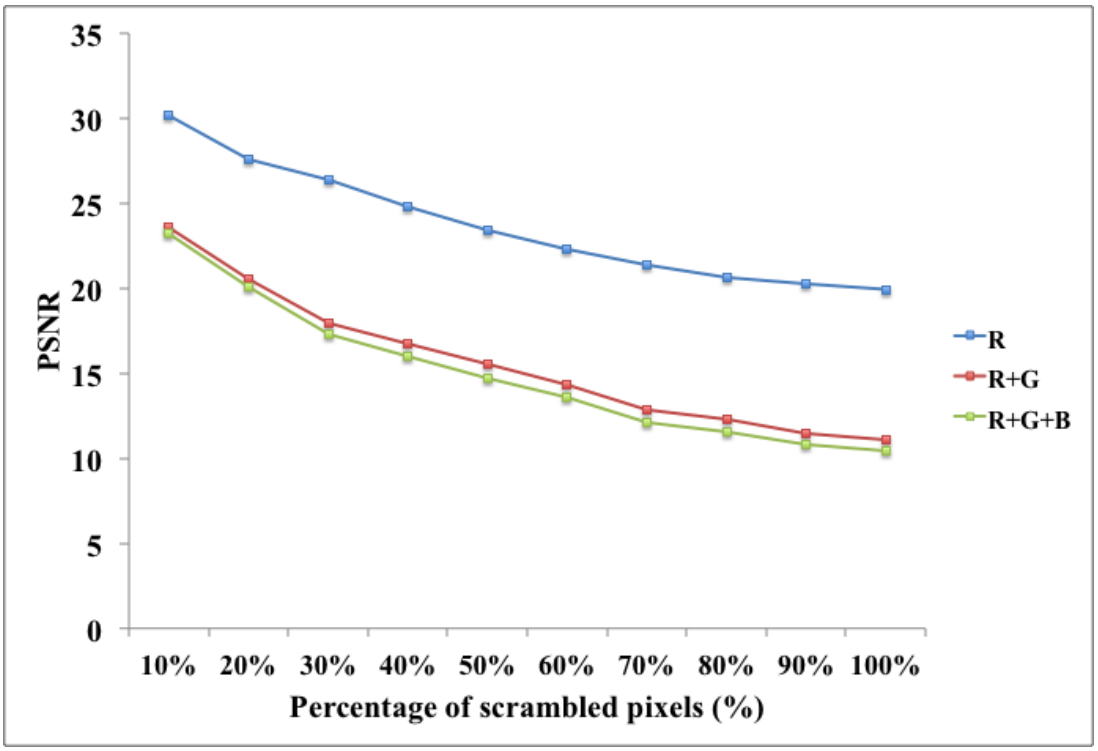


Figure 3.23: PSNR values of "kodim18" with 256 colors obtained by scrambling (R), (R and G), and (R, G, and B) components.

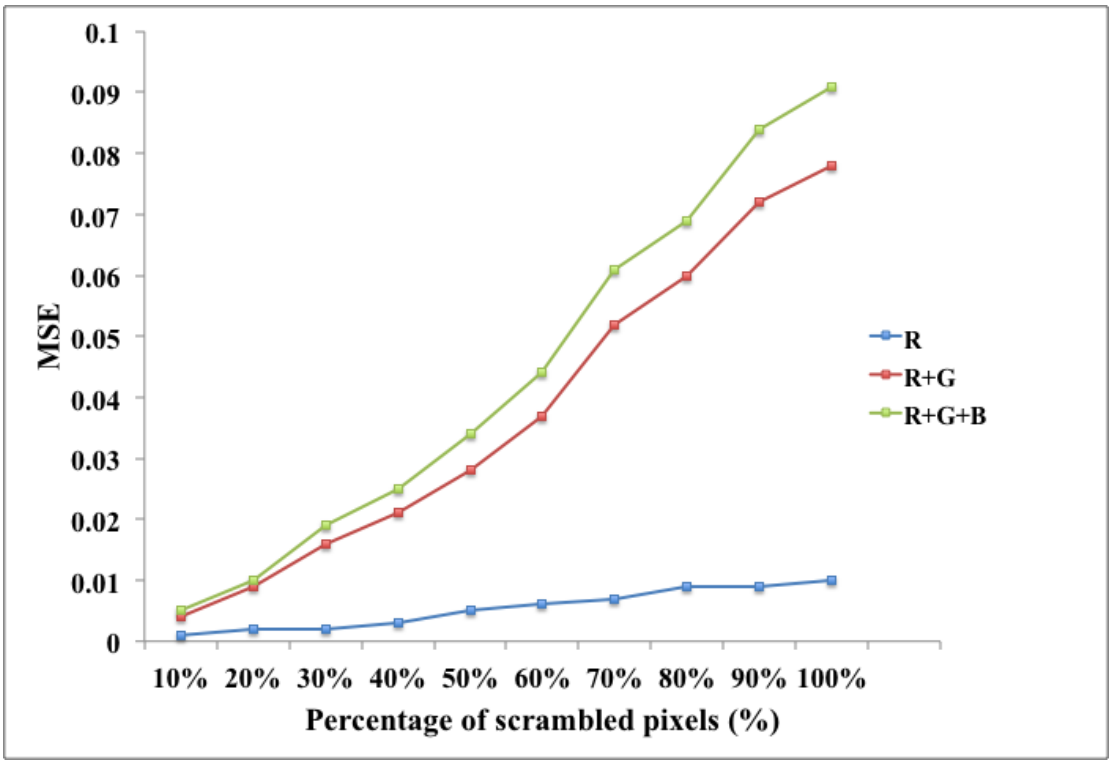


Figure 3.24: MSE values of "kodim18" with 256 colors obtained by scrambling (R), (R and G), and (R, G, and B) components.

Table 3.1: SSIM "kodim18" with 256 colors obtained by scrambling R, G, and B components.

	SSIM		
Percentage of scrambled pixels (%)	Prop.	Conv [19]	Conv [20]
10%	0.844	0.915	0.861
20%	0.635	0.683	0.749
30%	0.436	0.537	0.697
40%	0.327	0.386	0.573
50%	0.239	0.318	0.509
60%	0.139	0.204	0.475
70%	0.082	0.160	0.413
80%	0.049	0.126	0.354
90%	0.010	0.085	0.313
100%	0.006	0.082	0.311

Table 3.2: PSNR "kodim18" with 256 colors obtained by scrambling R, G, and B components.

	PSNR		
Percentage of scrambled pixels (%)	Prop.	Conv [19]	Conv [20]
10%	23.192	26.072	24.658
20%	20.088	20.978	22.659
30%	17.267	19.360	21.985
40%	15.962	16.999	20.303
50%	14.697	15.913	19.855
60%	13.573	14.504	19.721
70%	12.149	13.612	19.332
80%	11.588	13.169	19.123
90%	10.769	12.465	18.914
100%	10.407	12.220	18.880

Table 3.3: MSE "kodim18" with 256 colors obtained by scrambling R, G, and B components.

	MSE		
Percentage of scrambled pixels (%)	Prop.	Conv [19]	Conv [20]
10%	0.005	0.002	0.003
20%	0.010	0.008	0.005
30%	0.019	0.012	0.006
40%	0.025	0.020	0.009
50%	0.034	0.026	0.010
60%	0.044	0.035	0.011
70%	0.061	0.044	0.012
80%	0.069	0.048	0.012
90%	0.084	0.057	0.013
100%	0.091	0.060	0.013

3.5 Summary

The author proposed a new algorithm for a hierarchical scrambling method for palette-based images using bitwise operation. The proposed method scrambles the images more than the conventional methods [19, 20]. It also strengthens 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 the proposed method, whereas there

is a restriction on changing colors from the full colors in the conventional method. Therefore, the proposed method is more secure against unauthorized decryption as compared to the conventional methods. In addition, it can control the quality of scrambled images by using a single managed key. The future work involves studying different additional parameters to develop a more robust platform for effectively controlling and sharing of the target images.

Chapter 4

Integrated Model of Image Protection Techniques

Due to the development of digital communication technologies, there are various services, such as digital diagnosis, E-learning, and so on. Therefore, the digital contents should be secured properly because they can be easily manipulated and have a problems with copyright, authentication, data security, etc. As mentioned in Chapter 1, data hiding can be categorized into two types, i.e., Irreversible Data Hiding (IDH) and Reversible Data Hiding (RDH). The original image cannot be completely recovered in IDH. On the other hand, RDH is also known as invertible or lossless data hiding, and has been studied extensively to embed secret message bits into a cover object, such as an image/video or audio. In RDH, both the embedded message and the cover image should be losslessly recovered. Thus, these methods are desirable in some special scenarios, where no permanent change is accepted. In general, most of the RDH methods aim to provide the high capacity of the embedded data, the high resilience against possible attacks, the high quality of the embedded image, and the low computational complexity.

Histogram Shifting (HS) is one of the popular methods for RDH. The algorithm proposed by Ni et al. [36] is based on the HS method, in which the data is embedded to the peak of the histogram of an original image.

The Block-Permutation-Based Encryption (BPBE) schemes are one of the perceptual encryption techniques. In BPBE schemes, the original image is first divided into the definite block size, and then the four processes, that is, positional

scramble, block rotation/inversion, negative-positive transformation, and color component shuffling, are performed. The main feature of the BPBE schemes is that the compression efficiency of the encrypted images is compatible with an international standard such as JPEG compression [48]. The BPBE schemes have been proposed for the ETC technique, where a user securely transmits images through the SNSs provider.

Recently, the encryption based RDH using adaptive code embedding has been proposed [56]. This method has the advantage of maximum embedding rate of 1.72 bpp as well as the degradation of the final encrypted image. However, the drawback of this method is that the decryption process is not possible without the extraction of embedded data. Another drawback is that the access rights cannot be controlled according to different permission levels. Additionally, there is no consideration for the compatibility on compression efficiency using international standards. Therefore, the author proposes an integrated model of BPBE and RDH in this chapter [22, 58]. This method has the advantage of the image decryption without the extraction of embedded data. This method can also control the quality of an embedded image, and thus the access rights can be controlled according to various permission levels. The proposed method also considers to maintain the compression efficiency by using the international standards for lossless image compression, such as JPEG-LS. This method can be attractive in the scenario, such as doctor-nurse in a hospital, hierarchical file systems, and large organizations, where there is a hierarchical access control according to the different access rights. In some organizations, there is a complex hierarchy between the front line employees and the CEO. Different employees have to access different types of information according to their requirements. For instance, CEO is only the user with full permission. A manager may have the partial permission to know the salary of the employees but not personal information, such as telephone number. Furthermore, the author also proposes an efficient key derivation scheme to manage the multiple keys, which has been utilized in BPBE and RDH process. The experimental outcomes and analysis show the effectiveness of the proposed method.

4.1 Preliminary

4.1.1 BPBE scheme

The procedure for the BPBE scheme [50] is elaborated as follows.

Step 1 Divide each color component of color image $I = \{I_R, I_G, I_B\}$ into multiple blocks with $B_x \times B_y$ pixels.

Step 2 Permute the positions of the divided blocks randomly using key K_1.

Step 3 Rotate and invert each divided blocks using keys K_2 and K_3.

Step 4 Apply negative-positive transformation to the blocks using key K_4.

Step 5 Shuffle each color components in each block using key K_5.

Keys K_1, K_2, K_3, and K_4 are commonly used for the Red (R), Green (G), and Blue (B) color components.

4.1.2 RDH

As described in Section 2.4, the author has implemented the conventional RDH algorithm, which is employed in a spatial domain and is based on HS [36] as one of the examples, to this research. This algorithm is selected to maintain the quality of the embedded image. However, the other RDH algorithms can also be employed in the proposed method.

4.2 Proposed method

The author proposes an integrated model of BPBE and RDH, which can be well suitable for the hierarchical access control system [22, 58]. The main reason of using a hierarchical system is that the embedded data can be extracted as well as the encrypted image can be decrypted according to the different permission levels. Additionally, the security level can be controlled by embedding more confidential

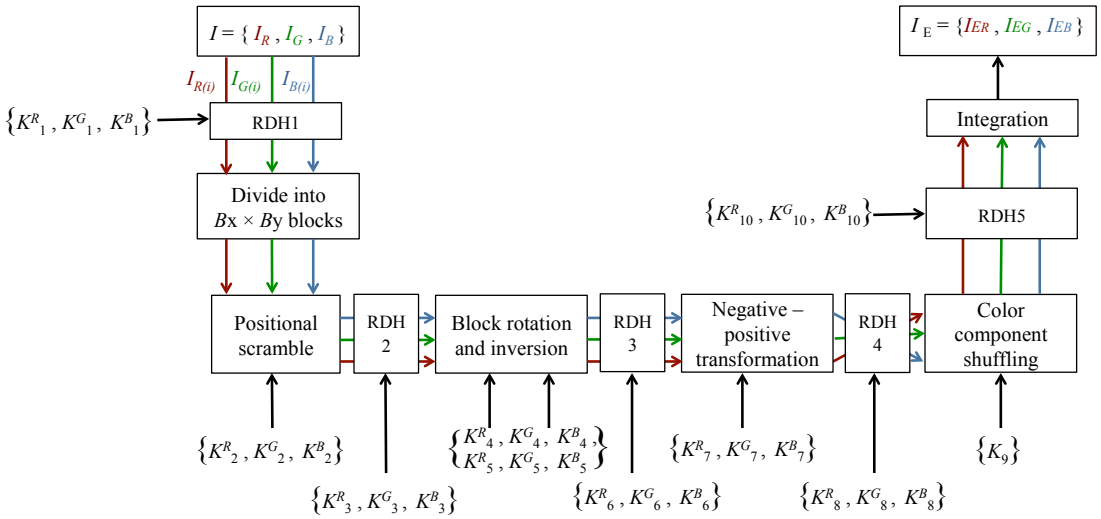


Figure 4.1: Encryption and embedding process.

data to an upper level, and less confidential data to a lower level in the hierarchical system. Therefore, the users with the higher permission level are permitted to extract more confidential data as compared to the user with the low permission level. In the case of decryption only permission, the users are able to decrypt the image, but are restricted to extract the embedded data. In this manner, the access control can be made flexible for the different users who are accessing the confidential data from various levels in the hierarchy.

The author utilizes three independent keys for R, G, and B components [47] for the encryption and the embedding process. It is possible to maintain the compression efficiency when we use JPEG-LS [64], which processes the images in RGB color space without conversion to any other color spaces.

4.2.1 Encryption and embedding process

In this section, the author elaborates the encryption and embedding process as shown in Fig. 4.1. For the simulation, the author has used a hundred different test images with 512 × 768 pixels (30) and 768 × 512 pixels (70) from the "Content-based image retrieval database [68]". The size of divided block is selected as 16 × 16 pixels for encryption.

Step 1 Apply RDH to original image $I = \{I_R, I_G, I_B\}$ of $M \times N$ pixels using keys K_1^R, K_1^G, and K_1^B.

Step 2 Divide each color component of the original image into multiple blocks with $B_x \times B_y$ pixels.

Step 3 Permute the positions of the divided blocks randomly using keys K_2^R, K_2^G, and K_2^B.

Step 4 Apply RDH using keys K_3^R, K_3^G, and K_3^B.

Step 5 Rotate and invert each block randomly using keys K_4^R, K_4^G, K_4^B, K_5^R, K_5^G, and K_5^B.

Step 6 Apply RDH using keys K_6^R, K_6^G, and K_6^B.

Step 7 Apply negative-positive transformation for each block using keys K_7^R, K_7^G, and K_7^B.

Step 8 Apply RDH using keys K_8^R, K_8^G, and K_8^B.

Step 9 Shuffle the three color components, i.e., R, G, and B in each block by using key K_9.

Step 10 Apply RDH using keys K_{10}^R, K_{10}^G, and K_{10}^B.

Step 11 Generate encrypted image $I_E = \{I_{ER}, I_{EG}, I_{EB}\}$ by integrating all the transformed blocks.

4.2.2 Key derivation

Due to the use of independent keys K_1^i, K_2^i, ..., K_8^i, K_{10}^i $(i = R, G, B)$ for three color components, a huge amount of keys would be generated in the proposed method. Therefore, the proper management of those multiple keys is a major issue. The author considers to derive an efficient key management scheme with the use of hash chains as well as decrease the number of managed keys [25]. The author assigns the derived keys to each step of the encryption and the embedding

processes. The number of managed keys are reduced to one, that is, key K_M. Keys K_x can be represented by

$$K_x = H^x(K_M), \tag{4.1}$$

where $x = 1, 2, \ldots, 10$ and $H(\cdot)$ is a one-way hash function.

Figure 4.2 shows the outline of the efficient key derivation scheme. Keys $K^i_{\mathrm{c}(u)}$ are the representation for the encryption process, whereas keys $K^i_{\mathrm{e}(u)}$ are the representation for the embedding process. In this case, $u (u = 1, 2, \ldots, 5)$ specifies the number of the encryption or the embedding process. A key $K^R_{\mathrm{e}(1)}$ can be derived by performing a one-way hash chain to the result obtained by XOR operation between key K_M and its associated random numbers a^R_{e} in embedding process. $K^G_{\mathrm{e}(1)}$ can be derived by a one-way hash function to the result of XOR operation between $K^R_{\mathrm{e}(1)}$ and a^G_{e}. Key $K^B_{\mathrm{e}(1)}$ can be obtained by performing a one-way hash function to the result obtained by XOR operation between $K^G_{\mathrm{e}(1)}$ and a^B_{e}. The key derivation for each color component can be given as below.

$$K^R_{\mathrm{e}(1)} = H(K_M \oplus a^R_{\mathrm{e}}), \tag{4.2}$$

$$K^G_{\mathrm{e}(1)} = H(K^R_{\mathrm{e}(1)} \oplus a^G_{\mathrm{e}}), \tag{4.3}$$

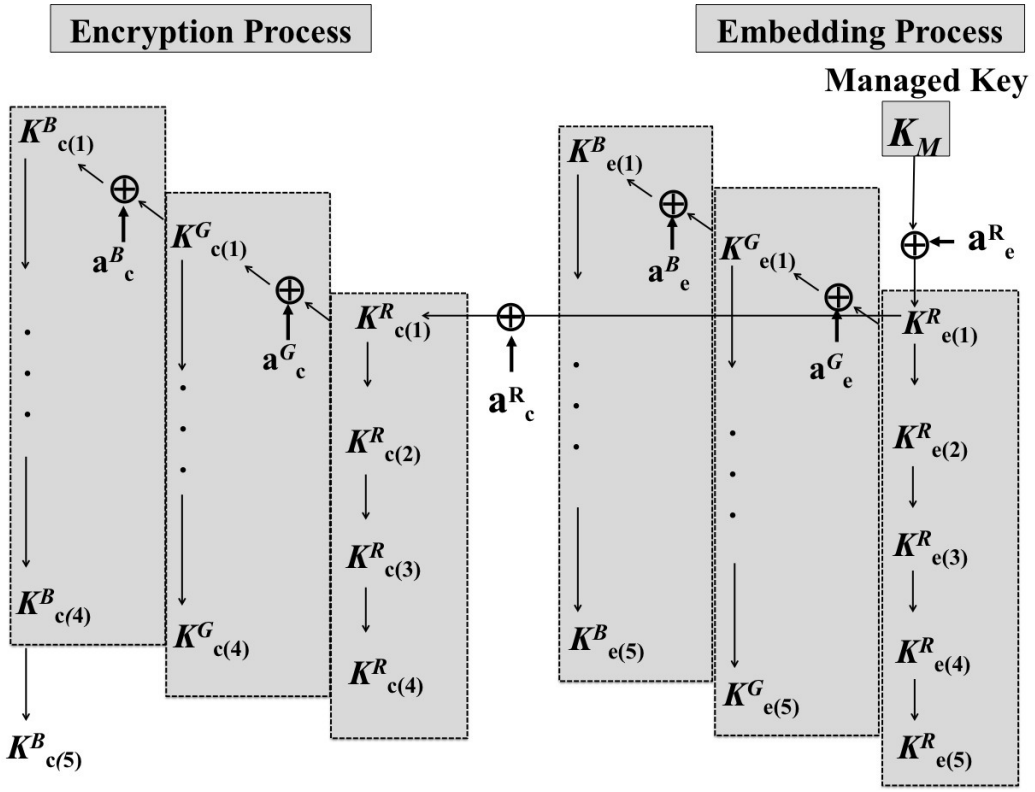
$$K^B_{\mathrm{e}(1)} = H(K^G_{\mathrm{e}(1)} \oplus a^B_{\mathrm{e}}), \tag{4.4}$$

where $\oplus$ represents a bitwise XOR operation.

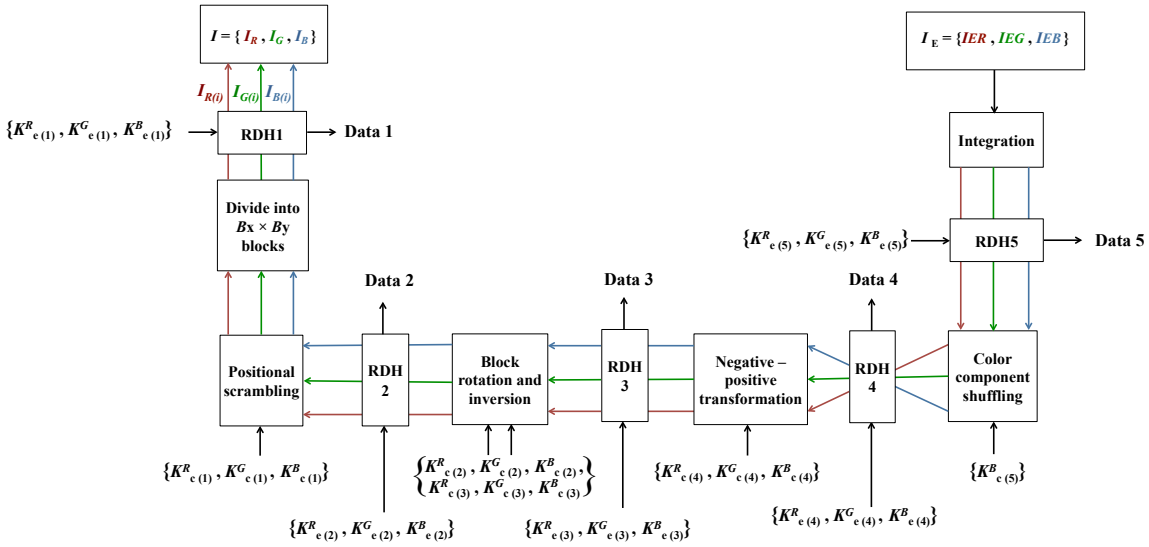
Additionally, by using hash chains, all other keys $K^i_{\mathrm{e}(1)}$, $K^i_{\mathrm{e}(2)}$, $\ldots$, $K^i_{\mathrm{e}(5)}$, $(i = R, G, B)$ can be derived as follows.

$$K^R_{\mathrm{e}(u)} = H^{u-1}(K^R_{\mathrm{e}(1)}), \tag{4.5}$$

$$K^G_{\mathrm{e}(u)} = H^{u-1}(K^G_{\mathrm{e}(1)}), \tag{4.6}$$



(a) Key derivation scheme.



(b) Decryption extraction process.

Figure 4.2: Key derivation.

$$K^B_{\mathrm{e}(u)} = H^{u-1}(K^B_{\mathrm{e}(1)}), \tag{4.7}$$

where $u = 2, 3, 4, 5$.

In encryption process, $K^R_{\mathrm{c}(1)}$ can be achieved by the result of XOR operation between $K^R_{\mathrm{e}(1)}$ and a^R_{c}. The key derivation process is described as below.

$$K^R_{\mathrm{c}(1)} = H(K^R_{\mathrm{e}(1)} \oplus a^R_{\mathrm{c}}), \tag{4.8}$$

$$K^G_{\mathrm{c}(1)} = H(K^R_{\mathrm{c}(1)} \oplus a^G_{\mathrm{c}}), \tag{4.9}$$

$$K^B_{\mathrm{c}(1)} = H(K^G_{\mathrm{c}(1)} \oplus a^B_{\mathrm{c}}), \tag{4.10}$$

Similarly,

$$K^R_{\mathrm{c}(u)} = H^{u-1}(K^R_{\mathrm{c}(1)}), \tag{4.11}$$

$$K^G_{\mathrm{c}(u)} = H^{u-1}(K^G_{\mathrm{c}(1)}), \tag{4.12}$$

$$K^B_{\mathrm{c}(u)} = H^{u-1}(K^B_{\mathrm{c}(1)}), \tag{4.13}$$

where $u = 2, 3, 4$.

Note that the key for the color component shuffling is single for each image. Hence, it is derived by

$$K^B_{\mathrm{c}(5)} = H(K^B_{\mathrm{c}(4)}). \tag{4.14}$$

Table 4.1: Embedding capacity at each level of Japan Image32.

	Embedding capacity (bits)	
Level	Single embedding	Double embedding
Data 1	18,709	37,233
Data 2	18,524	36,866
Data 3	18,476	36,090
Data 4	16,827	31,934
Data 5	15,297	27,396
Total	87,833	169,519

4.2.3 Decryption and extraction process

As depicted in Fig. 4.2, there are various hierarchical levels. Thus, the access rights for each hierarchical level can be controlled for the decryption and the extraction. The user with high permission is allowed to extract and decrypt more confidential data and images, respectively, than the user with low permission.

Table 4.1 represents the total embedding capacity of Japan Image32 at each level for single embedding (one time) and double embedding (two times). Tables 4.2 and 4.3 show the total embedding capacity and the PSNR values of test images for the single embedding and the double embedding, respectively. Figures 4.3 and 4.4 illustrate the simulation results of Japan Image32 and Japan Image22 obtained by various permissions for single embedding. The description of the decryption and extraction process for different permissions is given as follows.

4.2.3.1 Full permission

We assume that a user has the full access right to extract all the embedded data and to entirely decrypt the images. Therefore, the user has the full permission to obtain managed key K_M as shown in Fig. 4.2(a). If a user would obtain key K_M, the user is able to derive all twenty eight keys and retrieve the original image from the final encrypted image as depicted in Figs. 4.3(b) and 4.4(b). The user can also extract all the embedded data as represented in Table 4.1.

4.2.3.2 Partial permission

Let us suppose that another user is only allowed to extract Data 3, 4, and 5, as given in Fig. 4.2(b). Then, the user would obtain six keys, i.e., $K^R_{\mathrm{e}(3)}$, $K^G_{\mathrm{e}(3)}$, $K^B_{\mathrm{e}(3)}$, $K^R_{\mathrm{c}(4)}$, $K^G_{\mathrm{c}(4)}$, and $K^B_{\mathrm{c}(4)}$. Consequently, the user is able to derive the seven keys, i.e., $K^R_{\mathrm{e}(4)}$, $K^R_{\mathrm{e}(5)}$, $K^G_{\mathrm{e}(4)}$, $K^G_{\mathrm{e}(5)}$, $K^B_{\mathrm{e}(4)}$, $K^B_{\mathrm{e}(5)}$, $K^B_{\mathrm{c}(5)}$, extract Data 3, 4, and 5, and obtain the half encrypted image as shown in Figs. 4.3(c) and 4.4(c).

4.2.3.3 Decryption only permission

In this case, let us assume that a user is only permitted to completely decrypt the image, but is restricted in extracting the embedded data. When the user would obtain a key, i.e., $K^R_{\mathrm{c}(1)}$, then he is able to derive twelve keys, i.e., $K^G_{\mathrm{c}(1)}$, $K^B_{\mathrm{c}(1)}$, $K^R_{\mathrm{c}(2)}$, $K^G_{\mathrm{c}(2)}$, $K^B_{\mathrm{c}(2)}$, $K^R_{\mathrm{c}(3)}$, $K^G_{\mathrm{c}(3)}$, $K^B_{\mathrm{c}(3)}$, $K^R_{\mathrm{c}(4)}$, $K^G_{\mathrm{c}(4)}$, $K^B_{\mathrm{c}(4)}$, and $K^B_{\mathrm{c}(5)}$. Figures 4.3(d) and 4.4(d) represent the simulation results for decryption only permission. From the experimental results of a hundred images, the maximum and minimum values of PSNR are 43.93 dB (Japan Image27) and 37.42 dB (Indonesia Image35), respectively, as given in Table 4.2. Therefore, there is approximately only 6.51 dB of variation in PSNR values. The maximum embedding capacity for Iran Image13 is 525,427 bits with its corresponding PSNR value as 38.29 dB. Similarly, the minimum embedding capacity for Japan Image26 is 38,418 bits with its corresponding PSNR value as 38.81 dB. For this case, the proposed method is effective for Iran Image13 because it has comparatively higher embedding capacity than Japan Image26.

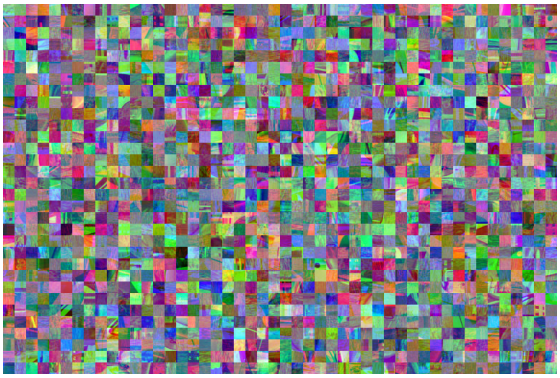
Table 4.2: Total embedding capacity (bits) and PSNR (dB) values for single embedding.

Image	Single embedding	
(768 × 512 pixels)	Total embedding capacity (bits)	PSNR(dB)
Japan Image22	41,576	38.34
Japan Image27	78,682	43.93 (max)
Japan Image32	87,833	40.41
Australia Image01	75,360	43.80
Australia Image03	86,720	41.69
Australia Image05	41,965	38.75
Indonesia Image01	285,821	37.91
Indonesia Image35	240,071	37.42(min)
Iran Image13	525,427(max)	38.29
Iran Image49	132,792	38.41
(512 × 768 pixels)	Total embedding capacity (bits)	PSNR(dB)
Japan Image26	38,418 (min)	38.81
Japan Image31	55,594	39.61
Australia Image02	41,212	42.77
Australia Image06	65,859	40.55
Indonesia Image20	89,424	40.98
Indonesia Image26	112,831	39.69
Iran Image10	218,532	42.29
Iran Image15	62,670	41.94

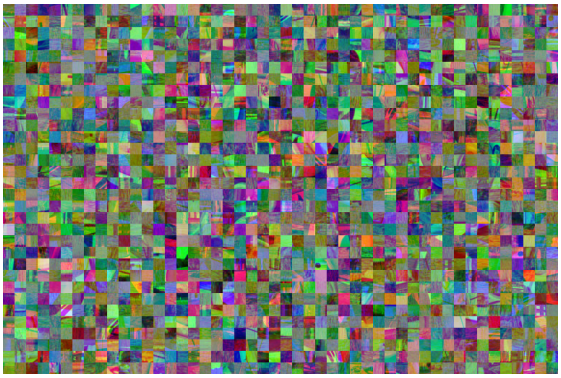
Table 4.3: Total embedding capacity (bits) and PSNR (dB) values for double embedding (Japan).

	Double embedding	
Image (768 × 512 pixels)	Total embedding capacity (bits)	PSNR(dB)
Japan Image01	131,529	37.44
Japan Image02	111,144	37.65
Japan Image08	90,016	34.53
Japan Image13	85,156	34.64
Japan Image15	88,180	32.93
Japan Image17	165,832	35.55
Japan Image20	101,275	35.66
Japan Image22	79,477	32.43
Japan Image27	142,136	39.68
Japan Image32	169,519	34.65

(a) Original image (Japan Image32).



(b) Final encrypted image.

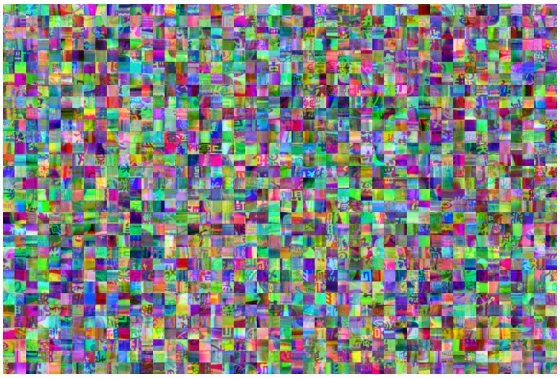


(c) Half encrypted image (Decryption: negative-positive transformation and color component shuffling, Extraction: Data 3, 4, and 5).

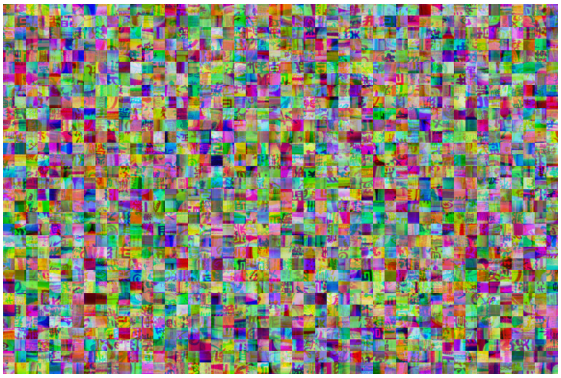
(d) Decryption only image.

Figure 4.3: Simulation results of Japan Image32 obtained by different permissions (single embedding).

(a) Original image (Japan Image22).



(b) Final encrypted image.



(c) Half encrypted image (Decryption: negative-positive transformation and color component shuffling, Extraction: Data 3, 4, and 5).

(d) Decryption only image.

Figure 4.4: Simulation results of Japan Image22 obtained by different permissions (single embedding).

4.3 Experimental results and analysis

4.3.1 Key space

Generally, there are different kinds of attacks on encryption, such as brute-force attacks, statistical attacks, differential attacks, and so on. A brute-force attack is a kind of trail-and-error methods, which is used to obtain the possible combination. In this section, the author determines the size of key space assuming the brute-force attacks. The conventional BPBE scheme [50] has four differ-

ent encryption processes, that is, positional scramble, block rotation/inversion, negative-positive transformation, and color component shuffling. It conducts the encryption with the identical keys for all three color components. The key space can be determined by the number of the divided blocks n. The four encryption processes are independent from each other. Therefore, the total key space is calculated by multiplying the key spaces of each encryption process as mentioned below.

When the original image with $M \times N$ pixels is divided into n blocks with $B_x \times B_y$ pixels, n is computed by

$$n = \frac{M \times N}{B_x \times B_y}. \tag{4.15}$$

Key space N_B is the number of permutation of n blocks in positional scramble, and is given by

$$N_B = n!. \tag{4.16}$$

Similarly, the combination of each four process of rotation and inversion generates some similar patterns. Thus, the maximum possible patterns generated by the rotation and the inversion processes are eight. Key space of the block rotation and inversion N_R is represented as

$$N_R = 8^n. \tag{4.17}$$

For the negative-positive transformation and the color component shuffling, the number of patterns N_N and N_C are two and six, respectively. Therefore, the total key space can be determined by

$$N_N = 2^n, \tag{4.18}$$

$$N_C = 6^n. \tag{4.19}$$

The total key space of the encrypted images N_A can be evaluated by

$$\begin{aligned} N_A &= N_B \times N_R \times N_N \times N_C \\ &= n! \times 8^n \times 2^n \times 6^n \\ &= n! \times 2^{5n} \times 3^n. \end{aligned} \tag{4.20}$$

The proposed method performs the encryption with the independent keys for all three color components. Key spaces $N_B^{'}$, $N_R^{'}$, and $N_N^{'}$ are calculated as

$$N_B^{'} = (n!)^3, \tag{4.21}$$

$$N_R^{'} = (8^n)^3, \tag{4.22}$$

$$N_N^{'} = (2^n)^3. \tag{4.23}$$

Hence, total key space of the encrypted image $N_A^{'}$ is evaluated by

$$\begin{aligned} N_A^{'} &= N_B^{'} \times N_R^{'} \times N_N^{'} \times N_C \\ &= (n!)^3 \times (8^n)^3 \times (2^n)^3 \times 6^n \\ &= (n!)^3 \times 2^{13n} \times 3^n. \end{aligned} \tag{4.24}$$

According to the above-mentioned analysis, the proposed method has a large key space than the conventional method [50] due to the use of independent keys for R, G, and B components in the encryption process. Although the conventional method [47] uses the independent keys for encryption, the key space of the proposed method is more complicated due to the embedding process. Thus, the proposed method is more secured by its large key space. In this way, the resilience against brute-force attacks can be improved.

4.3.2 Resilience against Jigsaw Puzzle Solvers (JPSs)

JPS is an attack that uses the correlation between the large number of pieces to retreive the original image. The encrypted image produced by the proposed method is composed of multiple blocks. Thus, it is required to analyze the security against JPSs. According to [69, 70], direct comparison D_c represents the ratio of the number of the pieces that are in the correct position. Neighbor comparison N_c is the ratio of the number of the correctly joined blocks. The largest component,

which is denoted as L_c, is the ratio of the number of the largest joined blocks having the correct adjacencies. As given in Table 4.4, the author has calculated the average scores of D_c, N_c, and L_c of seven different standard images, i.e., Lena, Mandrill, Milkdrop, Pepper, Girl, Lake, and Airplane of 512×512 pixels from "Signal and Image Processing Institute (SIPI) database [71]". The original image of 512×512 pixels are trimmed to 512×480 pixels to make a rectangular shape for JPSs analysis. It is noted that the block size for the encryption is 32×32 pixels. The scores of D_c, N_c, and L_c are $'1's$ when the image is completely assembled by JPSs, whereas these scores are $'0's$ if the puzzle are not assembled at all. Therefore, it is confirmed that the use of independent keys for encryption makes puzzle solvers more difficult to retrieve the original image compared to the use of identical keys.

4.3.3 Compression efficiency

The compression efficiency is evaluated by calculating the bitrate, which is given as

$$Bitrate(bpp) = \frac{Size\ of\ image\ file}{No.\ of\ pixels\ in\ original\ image}. \tag{4.25}$$

JPEG is a lossy compression algorithm. If we use lossy compression in the proposed method, the embedded data will be broken, and we cannot extract it. Thus, the lossless compression methods, such as JPEG-LS, is applied to the proposed method. As shown in Table 4.5, the compresssion performance of the encrypted image by the proposed method is not severely distorted as compared to that of the original image. Therefore, the proposed method is somehow compatible with JPEG-LS compression.

Table 4.4: Evaluation of JPSs using standard images with 512×512 pixels.

	BPBE scheme (32×32)	
Component	Identical	Independent
D_c(Avg)	0.102	0.002
N_c(Avg)	0.152	0.006
L_c(Avg)	0.197	0.008

Table 4.5: Calculation of bitrate after JPEG-LS compression (Iran Image13).

Image	Bitrate (bpp)
Original	13.10
Proposed method (independent)	13.73
Proposed method (identical)	13.70

4.4 Summary

The author proposed an integrated model of the image protection techniques in this chapter. The proposed method enables the hierarchical process for the encryption and the data embedding. This method is appropriate for the hierarchical access control system, where the permission is assigned according to the different access rights. The proposed method also considers to maintain the compression efficiency by using international standards for lossless image compression, such as JPEG-LS. Furthermore, the author also derives an efficient key derivation scheme to manage a large number of keys that are generated in the encryption and the data embedding processes. The size of key space is larger in the proposed method than the conventional method due to the use of independent keys and the embedding process. Therefore, the proposed method is more resilient against brute-force attacks. In addition, it is almost impossible for the present JPSs to illegally retrieve the original images.

Chapter 5

Conclusions

In this dissertation, the author proposed a study on hierarchical protection for copyrights of digital images. The main objective of this method is to control the access rights of various users at different permission levels by using different quality images. The author uses the hierarchical encryption and the integration of the hierarchical encryption and embedding process as described below.

Firstly, the author proposed a hierarchical scrambling scheme for palette-based images using bitwise operation. This method generates the scrambled images with different quality. The significant parameters, i.e., target pixels and target color components play an important role in evaluating the performance analysis. For this method, the target pixels for scrambling are taken by using pseudo-random numbers. Then, bitwise exclusive-OR is applied to concatenate the target pixel values and the corresponding pseudo-random numbers to manipulate the original pixel values. The colors in an image can be changed to any color from the 24-bit colors by this method. However, there is a restriction on changing colors from the full colors in the conventional methods [19, 20]. Therefore, the images in the proposed method are more distorted as compared to the conventional methods. Additionally, the author also introduces a hierarchical key assignment scheme to control the various access rights. The author has analyzed the experimental results for both the proposed and the conventional methods. The experimental results demonstrate that the proposed method is superior to the conventional methods [19, 20].

Secondly, the author proposed an integrated model of BPBE and RDH. As

mentioned earlier, BPBE has four processes for encryption, i.e., positional scramble, block rotation/inversion, negative-positive transformation, and the color component shuffling. HS is adopted for RDH. The proposed method allows a hierarchical process for the encryption and the embedding. Hence, it can be suitable for the hierarchical access control system, where the permission is assigned according to different access rights. A large number of keys are generated in the embedding and encryption process. Thus, the author derived an efficient key derivation scheme for the proper management of those multiple keys. The effective key derivation scheme provides the security according to the various acccess rights. In addition, the proposed method also considers to maintain the compression efficiency by using international standards for lossless image compression, such as JPEG-LS. The size of key space is more by the use of independent keys and the embedding process. Therefore, this method is more resilient against brute-force attacks. Furthermore, it is almost impossible for the present JPSs to illegally retrieve the original images by the proposed method.

The future work includes embedding the data that are related to the original image. For instance, it could be possible to control the visibility of Region of Interests (ROIs) with the embedded data in a hierarchical manner.

References

[1] V.M. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques," Proc. IEEE International Conference on Industrial Informatics, pp.709–716, 2005.

[2] H.L. Jin, M. Fujiyoshi, Y. Seki, and H. Kiya, "A data hiding method for JPEG 2000 coded images using modulo arithmetic," Electronics and Communications in Japan (Part III: Fundamental Electronic Science), vol.90, no.7, pp.37–46, 2007.

[3] I.E. Ziedan, M.M. Fouad, and D.H. Salem, "Application of data encryption standard to bitmap and JPEG images," Proc. IEEE National Radio Science Conference, pp.C16–1, 2003.

[4] B.B. Zhu, M.D. Swanson, and S. Li, "Encryption and authentication for scalable multimedia: Current state of the art and challenges," Proc. Internet Multimedia Management Systems V, pp.157–171, 2004.

[5] H. Kiya, S. Imaizumi, and O. Watanabe, "Partial-scrambling of images encoded using JPEG2000 without generating marker codes," Proc. IEEE International Conference on Image Processing, pp.III205–III208, 2003.

[6] J.M. Rodrigues, W. Puech, and A.G. Bors, "Selective encryption of human skin in JPEG images," Proc. IEEE International Conference on Image Processing, pp.1981–1984, 2006.

[7] A.K. Yekkala, N. Udupa, N. Bussa, and C.V. Madhavan, "Lightweight encryption for images," Proc. IEEE International Conference on Consumer Electronics, pp.1–2, 2007.

[8] A. Said, "Measuring the strength of partial encryption schemes," Proc. IEEE International Conference on Image Processing, pp.II:1126–1129, 2005.

[9] M. Fujiyoshi, S. Imaizumi, and K. Hitoshi, "Encryption of composite multimedia contents for access control," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E90–A, no.3, pp.590–596, 2007.

[10] S. Imaizumi and Y. Tobe, "Flexible partial encryption for palette-based images," Proc. International Workshop on Advanced Image Technology, 2015.

[11] C. Peng, R.H. Deng, Y. Wu, and W. Shao, "A flexible and scalable authentication scheme for JPEG2000 image codestreams," Proc. Eleventh ACM International Conference on Multimedia, pp.433–441, 2003.

[12] H.H. Yu, "Scalable encryption for multimedia content access control," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp.417–420, 2003.

[13] S. Imaizumi, O. Watanabe, M. Fujiyoshi, and H. Kiya, "Generalized hierarchical encryption of JPEG 2000 codestreams for access control," Proc. IEEE International Conference on Image Processing, pp.1094–1097, 2005.

[14] I. Ito and H. Kiya, "A new class of image registration for guaranteeing secure data management," Proc. IEEE Fifteenth International Conference on Image Processing, pp.269–272, 2008.

[15] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video," IEEE Transactions on Multimedia, vol.5, no.1, pp.118–129, 2003.

[16] I. Ito and H. Kiya, "One-time key based phase scrambling for phase-only correlation between visually protected images," EURASIP Journal on Information Security, vol.2009, no.1, p.3, 2009.

[17] Z. Tang, X. Zhang, and W. Lang, "Efficient image encryption with block shuffling and chaotic map," Multimedia Tools and Applications, vol.74, no.15, pp.5429–5448, 2015.

[18] D. Engel, T. Stütz, and A. Uhl, "A survey on JPEG2000 encryption," Multimedia Systems, vol.15, no.4, pp.243–270, 2009.

[19] A. Aryal, S. Imaizumi, and N. Aoki, "Hierarchical scrambling scheme for palette-based images," Proc. IEEE International Symposium on Intelligent Signal Processing and Communication Systems, pp.65–70, 2014.

[20] A. Aryal, S. Imaizumi, and N. Aoki, "Hierarchical scrambling for palette-based images using transposition cipher," Proc. International Conference on Advanced Imaging, pp.701–704, 2015.

[21] A. Aryal, S. Imaizumi, and T. Horiuchi, "Hierarchical Scrambling Method for Palette-Based Images Using Bitwise Operation," Bulletin of The Society of Scientific Photography of Japan, vol.26, no.1, pp.1–9, 2016.

[22] A. Aryal, S. Imaizumi, T. Horiuchi, and H. Kiya, "Integrated Model of Image Protection Techniques," Journal of Imaging, vol.4, no.1, pp.1–12, 2017.

[23] 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.

[24] S. Imaizumi, "A collusion-free key assignment scheme for hierarchical access control using recursive hash chains," Proc. IEEE International Symposium on Circuits and System, pp.445–448, 2013.

[25] S. Imaizumi, M. Fujiyoshi, and H. Kiya, "An efficient access control method for composite multimedia content," IEICE Electronics Express, vol.7, no.20, pp.1534–1538, 2010.

[26] J. Heikenfeld, P. Drzaic, J.S. Yeo, and T. Koch, "A critical review of the present and future prospects for electronic paper," Journal of the Society for Information Display, vol.19, no.2, pp.129–156, 2011.

[27] 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," Proc. Pacific-Rim Symposium on Image and Video Technology, pp.332–343, 2015.

[28] I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," IEEE Transactions on Image Processing, vol.6, no.12, pp.1673–1687, 1997.

[29] A.Z. Tirkel, C.F. Osborne, and R.G.V. Schyndel, "Image watermarking-a spread spectrum application," Proc. IEEE 4th International Symposium on Spread Spectrum Techniques and Applications, pp.785–789, 1996.

[30] M.M. Yeung and F.C. Mintzer, "Invisible watermarking for image verification," Journal of Electronic Imaging, vol.7, no.3, pp.578–591, 1998.

[31] M. Golijan, J.J. Fridrich, and R. Du, "Distortion-free data embedding for images," Proc. International Workshop on Information Hiding, pp.27–41, 2001.

[32] G. Xuan, J. Zhu, J. Chen, Y.Q. Shi, Z. Ni, and W. Su, "Distortionless data hiding based on integer wavelet transform," Electronic Letters, vol.38, no.25, pp.1646–1648, 2002.

[33] J. Tian, "Reversible data embedding using a difference expansion," IEEE Transactions on Circuits and Systems for Video Technology, vol.13, no.8, pp.890–896, 2003.

[34] D.M. Thodi and J.J. Rodriguez, "Expansion embedding techniques for reversible watermarking," IEEE Transactions on Image Processing, vol.16, no.3, pp.721–730, 2007.

[35] H.T. Wu, J.L. Dugelay, and Y.Q. Shi, "Reversible image data hiding with contrast enhancement," IEEE Signal Processing Letters, vol.22, no.1, pp.81–85, 2015.

[36] Z. Ni, Y.Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," IEEE Transactions on Circuits and Systems for Video Technology, vol.16, no.3, pp.354–362, 2006.

[37] X. Zhang, "Reversible data hiding in encrypted image," IEEE Signal Processing Letters, vol.18, no.4, pp.255–258, 2011.

[38] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," IEEE Transactions on Image Processing, vol.20, no.12, pp.3524–3533, 2011.

[39] X. Zhang, "Separable reversible data hiding in encrypted image," IEEE Transactions on Information Forensics and Security, vol.7, no.2, pp.826–832, 2012.

[40] P. Tsai, Y.C. Hu, and H.L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," Signal Processing, vol.89, no.6, pp.1129–1143, 2009.

[41] M.U. Celik, G. Sharma, A.M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," IEEE Transactions on Image Processing, vol.14, no.2, pp.253–266, 2005.

[42] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, "Reversible image watermarking using interpolation technique," IEEE Transactions on Information Forensics and Security, vol.5, no.1, pp.187–193, 2010.

[43] W.L. Tai, C.M. Yeh, and C.C. Chang, "Reversible data hiding based on histogram modification of pixel differences," IEEE Transactions on Circuits and Systems for Video technology, vol.19, no.6, pp.906–910, 2009.

[44] V. Sachnev, H.J. Kim, J. Nam, S. Suresh, and Y.Q. Shi, "Reversible watermarking algorithm using sorting and prediction," IEEE Transactions on Circuits and Systems for Video Technology, vol.19, no.7, pp.989–999, 2009.

[45] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding for all image formats," Security and Watermarking of Multimedia Contents IV, pp.572–584, 2002.

[46] C.C. Chang, C.C. Lin, and Y.H. Chen, "Reversible data-embedding scheme using differences between original and predicted pixel values," IET Information Security, vol.2, no.2, pp.35–46, 2008.

[47] S. Imaizumi, T. Ogasawara, and H. Kiya, "A Block-Permutation-Based-Encryption Scheme with Enhanced Color Scrambling," Proc. Scandinavian Conference on Image Analysis, LNCS, pp.562–573, 2017.

[48] K. Kurihara, S. Imaizumi, S. Shiota, and H. Kiya, "An Encryption-then-Compression System for Lossless Image Compression Standards," IEICE Transactions on Information and Systems, vol.E100-D, no.1, pp.52–56, 2017.

[49] O. Watanabe, A. Uchida, T. Fukuhara, and H. Kiya, "An Encryption-then-Compression System for JPEG 2000 Standard," Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, pp.1226–1230, 2015.

[50] K. Kurihara, M. Kikuchi, S. Imaizumi, S. Shiota, and H. Kiya, "An Encryption-then-Compression System for JPEG/Motion JPEG Standard," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E98-A, no.11, pp.2238–2245, 2015.

[51] J. Zhou, X. Liu, O.C. Au, and Y.Y. Tang, "Designing an efficient image encryption-then-compression system via prediction error clustering and random permutation," IEEE Transactions on Information Forensics and Security, vol.9, no.1, pp.39–50, 2014.

[52] Z. Erkin, A. Piva, S. Katzenbeisser, R.L. Lagendijk, J. Shokrollahi, G. Neven, and M. Barni, "Protection and retrieval of encrypted multimedia content: When cryptography meets signal processing," EURASIP Journal on Information Security, vol.2007, no.17, 2007.

[53] W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of encrypted grayscale images," IEEE Transactions on Image Processing, vol.19, no.4, pp.1097–1102, 2010.

[54] M. Johnson, P. Ishwar, V.P. D.Schnoberg, and K.Ramchandran, "On compressing encrypted data," IEEE Transactions on Signal Processing, vol.52, no.10, pp.2992–3006, 2004.

[55] R. Hu, X. Li, and B. Yang, "A new lossy compression scheme for encrypted gray-scale images," Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, pp.7387–7390, 2014.

[56] S. Yi and Y. Zhou, "Adaptive code embedding for reversible data hiding in encrypted images," Proc. IEEE International Conference on Image Processing, pp.4322–4326, 2017.

[57] T. Chuman, K. Kurihara, and H. Kiya, "Security evaluation for block scrambling-based ETC systems against extended jigsaw puzzle solver attacks," Proc. IEEE International Conference on Multimedia and Expo, pp.229–234, 2017.

[58] A. Aryal, S. Imaizumi, T. Horiuchi, and H. Kiya, "Integrated algorithm for block-permutation-based encryption with reversible data hiding," Proc. IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, pp.203–208, 2017.

[59] D. Taubman and M. Mercellin, "JPEG2000 image compression fundamentals, standards and practice," Kluwer Academic Publishers, 2002.

[60] M. Fallahpour and M.H. Sedaaghi, "High capacity lossless data hiding based on histogram modification," IEICE Electronice Express, vol.4, no.7, pp.205–210, 2007.

[61] Y. Izawa, S. Imaizumi, and H. Kiya, "A block-permutation-based image encryption allowing hierarchical decryption," Proc. IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, 2018. Accepted.

[62] 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.

[63] Ö. Küçük, "Design and analysis of cryptographic hash functions," Leuven: Katholieke Universiteit Leuven, 2012.

[64] "ISO/ICE 14495-1 : Information technology -Lossless and nearlossless compression of continuous-tone still images baseline," 1999.

[65] L.R. Knudsen and M.J. Robshaw, "Brute Force Attacks," in The Block Cipher Companion. Springer-Verlag, pp.95–108, 2011.

[66] "Secure hash standard FIPS PUB," National Institute of Standards and Technology, pp.180–184, 2002.

[67] "Kodak Lossless True Color Image Suite." `http://r0k.us/graphics/kodak/`.

[68] "Content-Based Image Retrieval Database." `http://imagedatabase.cs.washington.edu/groundtruth`.

[69] A. Gallagher, "Jigsaw puzzles with pieces of unknown orientation," Proc. IEEE Computer Vision and Pattern Recognition, pp.382–389, 2012.

[70] T.S. Cho, S. Avidan, and W.T. Freeman, "A probabilistic image jigsaw puzzle solvers," Proc. IEEE Computer Vision and Pattern Recognition, pp.183–190, 2010.

[71] "The USC-SIPI Image Database." `http://sipi.usc.edu/database`.