JPEG Image Encryption with Improved Format Compatibility and File Size Preservation

Junhui He, Shuhao Huang, Shaohua Tang, *Member, IEEE*, Jiwu Huang, *Fellow, IEEE*

***Abstract*—Image encryption techniques can be used to ensure the security and privacy of valuable images. The related works in this field have focused more on raster images than on compressed images. Many existing JPEG image encryption schemes are not quite well compatible with the JPEG standard, or the file size of an encrypted JPEG image is apparently increased. In this paper, a novel bitstream-based JPEG image encryption method is presented. First, the groups of successive DC codes that encode the quantized DC coefficient differences with the same sign are permuted within each group. Second, the left half and the right half of a group, whose size will increase with the number of iterations, of consecutive DC codes may be swapped with each other, depending on whether an overflow of quantized DC coefficients occurs during decoding. Third, all AC codes are classified into 63 categories according to their zero-run lengths, then the AC codes within each category are respectively scrambled. Finally, all MCUs except for DC codes are randomly shuffled as a whole. Moreover, an image-content-related encryption key is employed to provide further security. The experimental results show that the file size of an encrypted JPEG image is almost the same as that of the corresponding plaintext image except for slight variations because of byte alignment. Additionally, the quantized DC coefficients decoded from an encrypted JPEG image will not fall outside the valid range. Improved format compatibility is provided compared with other related methods. Moreover, it is unnecessary to perform entropy encoding again because all of the encryption operations are performed directly on the JPEG bitstream. The proposed method is proved to be secure against brute-force attacks, differential cryptanalysis, known plaintext attacks and outline attacks. Our proposed method can also be applied to color JPEG images.**

***Index Terms*—JPEG bitstream, image encryption, format compatibility, file size preservation**

I. Introduction

THE popularization of mobile terminals leads to an explosive growth of user-generated multimedia contents. The continuous increase in network bandwidth and the decreasing cost make it easy for users to backup and synchronize multimedia contents to public cloud servers. However, transmitting multimedia contents through public channels or saving them to cloud servers may face threats of illegal interception or unauthorized access. How to tighten the security and privacy of multimedia contents has become an important problem.

There are various types of formats for multimedia contents, among which image is a typical one. Encryption is an effective approach to securing valuable images, such as military images, blueprints, medical images and private photos, by making them incomprehensible. Many chaotic image encryption algorithms have been proposed in the literature, e.g., shuffling of image pixels using a three-dimensional chaotic cat map [1] and image encryption based on a simple 1D chaotic system [2]. Recently, a method of reversible image transformation is proposed to transform the content of an original image into the content of another target image so as not to arouse the public cloud servers' attention [3]. However, most of these methods are only suitable for raster images and cannot be directly extended to deal with compressed images.

DCT-based JPEG compression [4] is a classic method of lossy compression for digital images. Furthermore, JPEG is generally not only the default format of the images produced by digital cameras, scanners, and other image capture devices but also the most commonly used format for storing and transmitting photographic images on the Internet. Therefore, JPEG image encryption has attracted considerable attention. Though a number of different JPEG image encryption schemes have been proposed [5]–[25], there are various deficiencies in these methods. The detailed analysis of the weaknesses of the existing encryption techniques is presented in Section II.

In this paper, a novel bitstream-based JPEG image encryption method that can overcome the shortcomings analyzed in Section II is proposed. In addition to the groups of consecutive DC codes that encode the quantized DC coefficient differences with the same sign being scrambled within each group, the left half and the right half of a variable-size group of continuous DC codes are swapped with each other depending on whether an overflow of quantized DC coefficients results during JPEG decompression. Furthermore, all the AC codes are classified into 63 categories based on their zero-run lengths, and then the AC codes within the same category are randomly permuted. Finally, all MCUs except for DC codes are globally shuffled. In addition, an image-content-related key is employed to defeat known plaintext attacks.

The remainder of this paper is organized as follows. In Section II, the weaknesses of the existing JPEG encryption schemes are analyzed. Section III describes the proposed scheme in detail. The experimental results and analysis are presented in Section IV. The security of our scheme is discussed in Section V. Finally, the conclusions are stated in Section VI.

This work was supported in part by the National Natural Science Foundation of China (61632013, U1636202, 61332012) and in part by Shenzhen R&D Program (JCYJ20160328144421330).

J. He, S. Huang and S. Tang are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou Higher Education Mega Centre, Guangzhou 510006, China (e-mail: hejh@scut.edu.cn; cshuangsh@mail.scut.edu.cn; shtang@ieee.org).

J. Huang is with the Guangdong Key Laboratory of Intelligent Information Processing and Shenzhen Key Laboratory of Media Security, College of Information Engineering, Shenzhen University, Shenzhen, China (email: jwhuang@szu.edu.cn)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2817065, IEEE Transactions on Multimedia

II. Analysis of the Weaknesses of the Existing JPEG Image Encryption Techniques

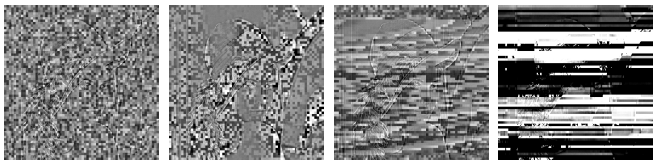
In this section, the respective weaknesses of the existing encryption techniques toward JPEG images are analyzed. Among the compressed image data of a JPEG image, the quantized DC coefficients, AC coefficients and table specifications are usually used for encryption.

A. Encryption of quantized DC coefficients

The DC coefficients frequently contain a significant fraction of the image energy and carry important visual information. If the DC coefficients are not encrypted, the outline of an image can be revealed. The known DC coefficient encryption techniques and their weaknesses are summarized in the following.

- Random permutation [9], [24]: The correlations between neighboring DC coefficients, which contribute substantially to the compression ratio, will be greatly weakened because of random permutation. Thus, the file size of an encrypted JPEG image will apparently increase.
- Category address mapping [24]: The mapping enlarges the difference between the quantized DC coefficient of the current block and that of the previous block of the same component, which will lead to a noticeable increase in the file size of an encrypted JPEG image.
- Region-basis permutation [12], [21]: The change in the file size introduced by region-basis permutation may not be significant for the pixel values within such a region are of subtle variation. However, the perceptual content of a plaintext JPEG image will be less distorted by region-basis permutation compared to random permutation, e.g., visual patterns can be observed in the encrypted JPEG images.
- Coefficient difference encryption [10], [11], [14], [15], [17], [19]–[21], [23]: The correlations between adjacent DC coefficients are exploited and the compression ratio will be retained. However, an overflow of quantized DC coefficients will occur during JPEG decoding.

To demonstrate the issues of the DC coefficient encryption techniques mentioned above, many experiments have been conducted, and the experimental results are presented in Fig. 1, Table I, Table II and Fig. 2. The images used here, i.e., Aerial, Airplane, Couple, Lena and Pepper, are downloaded from USC-SIPI image database and converted into JPEG format using OpenCV with a quality factor of 85. And the number of labels used in the region-basis permutation is 3. As shown in Fig. 1, the encryption technique based on random permutation gives the best visual effect, but it suffers a significant increase in file size, which is shown in Table I. Besides a noticeable file size increase, apparent patterns can be observed in the encrypted images generated using category address mapping, which indicates poor visual security. Moreover, as shown in Table II, coefficient difference encryption leads to a severe overflow of quantized DC coefficients during JPEG decompression, which will cause the decoded images from the same encrypted image to look very different, as shown in Fig. 2, when the corresponding encrypted image is opened in different application software. The significant difference in the decoded images may make it difficult for users to judge the visual security of the corresponding JPEG image encryption schemes.



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

Fig. 1: Encrypted images of Lena generated using (a) random permutation, (b) category address mapping, (c) region-basis permutation, and (d) coefficient difference encryption (opened in MATLAB).

TABLE I: Percentage of file size change resulting from DC encryption ('+' means an increase, and '−' means a decrease).

Encryption method	File size change (%)				
	Aerial	Airplane	Couple	Lena	Pepper
Random permutation	+1.039	+3.788	+2.642	+3.632	+3.637
Category address mapping	+2.422	+5.346	+2.111	+2.830	+3.035
Region-basis permutation	-0.050	-0.151	-0.022	+0.088	0
Coefficient difference encryption	-0.008	-0.002	+0.006	+0.004	-0.002

TABLE II: Percentage of the overflowed quantized DC.

Encryption method	DC outside the valid range (%)				
	Aerial	Airplane	Couple	Lena	Pepper
Random permutation	0	0	0	0	0
Category address mapping	5.47	11.08	1.05	2.71	4.25
Region-basis permutation	0	0	0	0	0
Coefficient difference encryption	86.18	89.26	98.80	86.06	92.33

(a) Pepper (b) In Photoshop (c) In MATLAB (d) In GIMP

Fig. 2: The same encrypted Pepper generated using DC coefficient difference encryption is opened in different software.

B. Encryption of quantized AC coefficients

Generally, the AC coefficients contain image detail information. If only the DC coefficients are encrypted, then a good approximation of the plaintext JPEG image may be obtained by assigning any valid value to the DC coefficients. Moreover, merely the unencrypted AC coefficients will reveal the edge information of an image. Hence, the AC coefficients are also needed to be encrypted. The typical AC coefficient encryption methods and their shortcomings are summarized below.

- Intra-block permutation [5]: The statistical characteristics of the AC coefficients in a block may be destroyed. Thus, the encryption method based on intra-block permutation may suffer from a significant increase in file size.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2817065, IEEE Transactions on Multimedia

TABLE III: Percentage of file size change resulting from AC encryption ('+' means an increase, and '−' means a decrease).

Encryption method	File size change (%)				
	Aerial	Airplane	Couple	Lena	Pepper
Intra-block permutation	+43.843	+66.518	+56.077	+64.942	+58.668
Intra-subsection permutation	+5.851	+7.161	+7.403	+7.275	+6.417
Appended bits encryption	+0.036	-0.009	+0.081	+0.080	-0.002
Sign encryption	-0.005	-0.029	+0.017	+0.020	-0.008
Category Address mapping	+0.025	-0.029	+0.044	+0.051	-0.008
Intra-block symbol permutation	+0.059	-0.009	+0.068	+0.020	0
Frequency-based permutation	+9.014	+17.112	+14.300	+13.417	+10.302
Block-based permutation	+0.005	-0.099	+0.037	+0.022	-0.060

- Intra-subsection permutation [6]: Although the impact of intra-subsection scrambling on the statistical distribution is somewhat lowered, the file size of an encrypted JPEG image will still increase noticeably.
- Non-zero coefficient encryption: The appended bits of AC codes are XOR-ed with a keystream [11], [13], [15], [17], [19]–[21], or only the signs of all non-zero quantized AC coefficients are encrypted [6], [12]. Besides, every non-zero AC coefficient is mapped to another value in the same category assigned to it [24]. These encryption techniques have no adverse effect on compression performance because the lengths of the appended bits will be preserved. However, some schemes are insecure against known plaintext attacks when an encryption key is used to encrypt multiple images.
- Intra-block symbol permutation [11]–[13], [15], [18], [23]: The outline attacks proposed in [9], [12] can be used to sketch the outline of an original JPEG image from the corresponding encrypted image.
- Inter-block symbol permutation [14], [21]: The scrambling may destroy the format compatibility because the number of AC coefficients in an 8×8 DCT block may exceed 63.
- Frequency-based permutation [9]: This method can successfully withstand the outline attacks, but it may decrease the image compression ratio.
- Block-based permutation [10]–[15], [18], [23], [24]: If all DCT blocks are randomly scrambled, then it is difficult to obtain the outline image using the existing outline attacks.

The experimental results of file size change and security against the outline attack based on the energy of AC coefficients (EAC) [12] are respectively illustrated in Table III and Fig. 3 to demonstrate the issues of the AC coefficient encryption techniques previously described. The images used here are the same as those used in Section II-A. And the number of subsections employed in the intra-subsection permutation is 4, as suggested in [6]. As shown in Table III, three of the AC coefficient encryption methods, including intra-block permutation, intra-subsection permutation and frequency-based permutation, suffer from a substantial increase in file size. In the outline attack experiments, each AC coefficient encryption technique is combined separately with random permutation of DC coefficients. As shown in Fig. 3, the outline of the original JPEG image has been successfully generated with EAC attack, except frequency-based and block-based permutation.

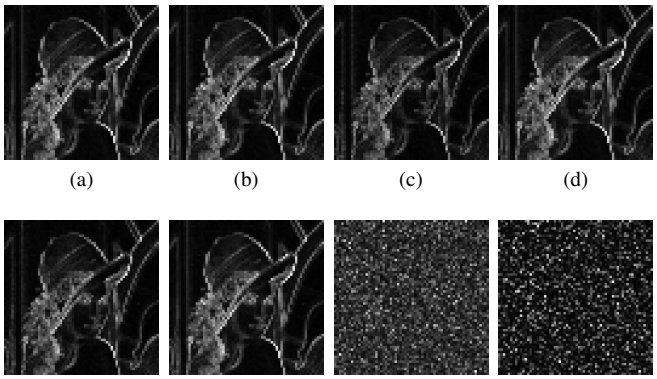


Fig. 3: Outline images obtained using EAC [12] from the encrypted images generated by: (a) intra-block permutation, (b) intra-subsection permutation, (c) appended bits encryption, (d) sign encryption, (e) category address mapping, (f) intra-block symbol permutation, (g) frequency-based permutation, and (h) block-based permutation.

C. Encryption of table specifications

In addition to DC and AC coefficients, table specifications are included as important parts within JPEG compressed image data. The table specifications may be modified for encryption. The existing approaches for altering the table specifications are shown below.

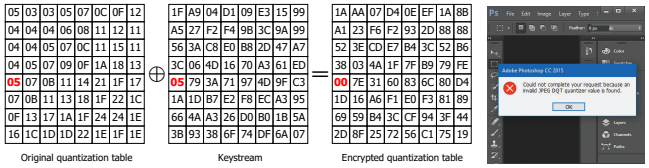
- Huffman table encryption [26]: Random permutation of the Huffman codeword list may be applied for image encryption. However, the file size of an encrypted JPEG image may increase because random permutation renders the employed Huffman table less than optimal. Moreover, the size of the permutation key space is limited.
- Quantization table encryption [17], [20], [23]: If a value in the quantization tables is encrypted to be invalid, then the format compatibility will be affected. Moreover, XOR-based quantization table encryption is insecure against known plaintext attacks.

As shown in Fig. 4, if the value in red of the quantization table is encrypted to 0, an error message will be displayed when we attempt to open the corresponding encrypted image using Photoshop. This result indicates that the encrypted JPEG image may be incompatible with some image application software. Moreover, if a plaintext JPEG image is encrypted only by XOR-ing its quantization table with a random keystream, the attackers may attempt to decrypt the encrypted image with the quantization table suggested in the JPEG standard [4].

III. Proposed Method

The in-depth discussion presented in Section II may lead to the conclusion that it is necessary for a JPEG encryption scheme to encrypt both the quantized DC and AC coefficients. Although four distinct modes, namely, baseline sequential, progressive, lossless and hierarchical, are defined in the JPEG standard, only the baseline sequential decoding process is required for all DCT-based decoding processes [4]. Therefore,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2817065, IEEE Transactions on Multimedia



(a) Quantization table encryption using XOR (b) Error msg.

Fig. 4: A zero value (red) in the encrypted quantization table resulting from XOR-based encryption causes an error message during JPEG decompression.

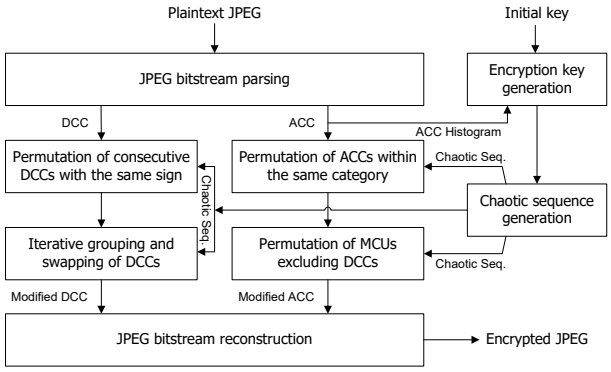


Fig. 5: Diagram of the proposed encryption method.

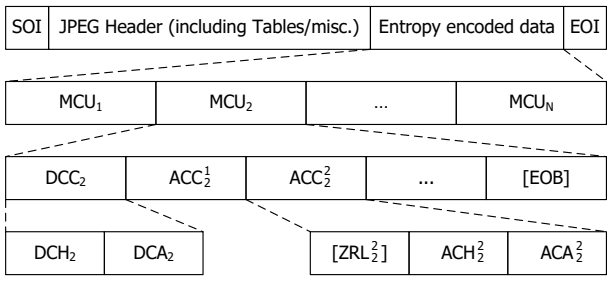


Fig. 6: Simplified syntax of sequential DCT-based JPEG ([·] means the element may be absent).

the following discussion will be limited to JPEG images encoded in the baseline sequential mode. A diagram of the proposed method is presented in Fig. 5. The detailed procedures in the diagram are described in the following subsections.

A. JPEG bitstream parsing

According to the JPEG format specified in [4], the syntax for the constituent parts of a grayscale JPEG image may be simplified as in Fig. 6. The JPEG bitstream begins with a start-of-image (SOI) marker, contains a JPEG header (including table specifications/misc.) and an entropy encoded data part, then ends with an end-of-image (EOI) marker.

The entropy encoded data comprises multiple minimum coded units (MCUs), each MCU consists of one DC code (DCC) for the DC coefficient, several AC codes (ACCs) for the AC coefficients and an end-of-block (EOB) code. If the last AC coefficient in the zig-zag sequence is not zero, the EOB code is bypassed. Let MCU_i $(i = 1, 2, \cdots, N)$ denote the i-th MCU, where N is the total number of non-overlapping DCT blocks. The DCC and ACCs in MCU_i are respectively represented by DCC_i and ACC_i^j $(j = 1, 2, \cdots, M_i)$, where M_i represents the number of non-zero AC coefficients in MCU_i. For an 8×8 DCT block, M_i lies in $[0, 63]$.

The DCC_i is composed of a Huffman code DCH_i and the appended bits DCA_i, which encodes the difference between the quantized DC coefficient of the current block and that of the previous block. And the ACC_i^j consists of zero or more consecutive ZRL_i^j, one Huffman code ACH_i^j and the appended bits ACA_i^j. Each ZRL_i^j represents 16 continuous zero coefficients, and ACA_i^j specifies the sign and exact amplitude of the corresponding non-zero AC coefficient. The number of zero AC coefficients preceding a non-zero AC coefficient, called zero-run length, is determined by the ZRL_i^j and ACH_i^j. If the zero-run length is less than 16, there is no ZRL_i^j in ACC_i^j.

Let the value encoded by the DCA_i be denoted as d_i and the corresponding quantized DC coefficient be represented by QD_i. The sign of d_i can be determined by the first bit of DCA_i. Specifically, if the first bit of DCA_i is 0, then d_i is negative; otherwise, it is non-negative. According to the DPCM encoding model for DC coefficients, we have

$$\text{QD}_i = \sum_{j=1}^{i} d_j, \text{ for } i = 1, 2, \cdots, N. \tag{1}$$

B. Random sequence generation

A random sequence will be essential for the permutation employed in our scheme. Due to their unpredictable behavior and high sensitivity to initial conditions, chaotic maps are generally used in the generation of random sequences. The logistic map is applied in the proposed method because of its simplicity. In fact, other chaotic sequence generation algorithms can be easily incorporated into our scheme. Mathematically, the logistic map is defined as

$$x_{n+1} = \mu x_n (1 - x_n) \tag{2}$$

where $x_n \in [0, 1.0]$ and $\mu \in [0, 4.0]$. When the initial value x_0 and the parameter μ are specified, the chaotic sequence $x_0, x_1, x_2 \cdots$ is determined. To make the logistic map have a chaotic behavior, the parameter μ should be carefully chosen, such as the values between 3.57 and 4 may be used.

To generate a random permutation sequence of length N, N chaotic numbers $x_1, x_2, \cdots, x_N$ are generated iteratively according to (2). If the sorted chaotic sequence by their values is $x_{i_1}, x_{i_2}, \cdots, x_{i_N}$, a permutation sequence $i_1, i_2, \cdots, i_N$ will be obtained as shown below.

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \\ 1 & 2 & \cdots & N \end{pmatrix} \xrightarrow{\text{Sort 1st row}} \begin{pmatrix} x_{i_1} & x_{i_2} & \cdots & x_{i_N} \\ i_1 & i_2 & \cdots & i_N \end{pmatrix} \tag{3}$$

C. Adaptive encryption key generation

To defeat known plaintext attacks, a method of adaptive encryption key generation is employed in our proposed scheme. Rather than using the same encryption key for many JPEG images, distinct keys are used for encrypting different JPEG images. Based on a user-chosen initial key $k_m = \{x_m, \mu_m\}$, the image-content-related encryption key $k_e = \{x_e, \mu_e\}$ is generated in the following manner.

As shown in Fig. 6, each MCU may contain a different number of ACCs. For an 8×8 DCT block, the number of ACCs in a MCU will not be larger than 63. Let h_i $(i = 0, 1, \cdots, 63)$ denote the number of blocks in which there are i ACCs. Since our proposed encryption algorithm will not change the number of ACCs in each MCU, h_i will remain intact after encryption. Thus, h_i can be recovered without error from the encrypted JPEG image.

All the indices i's and the corresponding number h_i's are input to a hash function to generate a code. SHA3-512 hash algorithm is applied in our scheme. The 512 bits of the hash code are equally partitioned into two parts—the left part and the right part, both of which are then divided into 29 groups. Each of the first 28 groups consists of 9 bits, and the remaining 4 bits are collected in the last group. For the left part, the numbers of binary '1's in the 29 groups, that is, 29 digits, are appended to x_m of the initial key k_m to generate x_e of the encryption key k_e. In a similar way, the numbers of binary '1's in the 29 groups of the right part are appended to μ_m of k_m to form μ_e of k_e. With the use of the encryption key $k_e = \{x_e, \mu_e\}$ as a seed for the logistic map defined in (2), the image-content-related chaotic sequences will be produced for the encryption of DCCs and ACCs. To implement our method with higher precision, GMP (GNU Multiple Precision Arithmetic Library) [27] is employed in our program codes.

D. Encryption of DCCs

For the encryption of DCCs, each group of the consecutive DCCs with the same sign is first internally permuted. Then, the left half and right half of a variable-size group of continuous DCCs may be swapped with each other iteratively.

1) Range of quantized DC coefficients: In the process of JPEG compression, an image will be first divided into 8×8 blocks, and the pixel values in every block shall be level shifted to a signed representation. Then each image block is transformed into frequency domain using the two-dimensional DCT defined in (4).

$$G_{uv} = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} g_{xy} \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \quad (4)$$

where u $(0 \le u \le 7)$ is the horizontal spatial frequency and v $(0 \le v \le 7)$ is the vertical spatial frequency, $C_u, C_v = \frac{1}{\sqrt{2}}$ for $u, v = 0$, otherwise $C_u, C_v = 1$. g_{xy} is the level shifted value at the coordinates (x, y), and G_{uv} is the DCT coefficient at the coordinates (u, v). By setting $u = 0$ and $v = 0$ in (4), the DC coefficient G_{00} can be calculated using:

$$G_{00} = \frac{1}{8} \sum\nolimits_{x=0}^{7} \sum\nolimits_{y=0}^{7} g_{xy} \quad (5)$$

For 8-bit images, the level shifted pixel values fall in the range $[-128, 127]$ after subtracting 128 from $[0, 255]$. According to (5), the range of DC coefficients can be obtained.

$$-1024 \le G_{00} \le 1016 \quad (6)$$

Therefore, the valid range of the quantized DC coefficients can be calculated as follows:

$$\text{round}\left(\frac{-1024}{Q_{00}}\right) \le \text{QD}_i \le \text{round}\left(\frac{1016}{Q_{00}}\right) \quad (7)$$

where Q_{00} is the quantization step for the coefficient at the coordinates $(0, 0)$, which is related to the quality factor of a JPEG image. To simplify the expression later in this paper, the lower and upper bounds of the quantized DC coefficients shown in (7) will be denoted as MIN and MAX, respectively. Thus, we can simplify (7) as

$$\text{MIN} \le \text{QD}_i \le \text{MAX} \quad (8)$$

From (1), we can observe that if the order of d_is or the values of d_is are altered during the encryption process, then the values of QD_is will also be changed. Therefore, the values of QD_is may fall outside the range specified in (8). The design of DC encryption should take this side effect into consideration.

2) Permutation of consecutive DCCs with the same sign: In order to facilitate the description of our algorithm, a DCC_i is considered to be non-negative or negative if the corresponding d_i is non-negative or negative. The method for quickly judging the sign of d_i was described in Section III-A. In the following discussion, the non-negative or negative DCC_i will be denoted by DCC_i^+ or DCC_i^- respectively.

All the N DCCs are collected together in the encoding order, and every segment of consecutive DCCs with the same sign forms into a group. Then, many groups of DCCs with the same sign will be produced. Let M and S_t denote the total number of groups and the t-th group, respectively, and $|S_t|$ denotes the number of DCCs in the group S_t. Clearly, $|S_1| + |S_2| + \cdots + |S_M| = N$. Without loss of generality, the M groups may be described as follows:

$$\begin{aligned} &\underbrace{\text{DCC}_1^+, \cdots, \text{DCC}_{n_1}^+}_{S_1}, \underbrace{\text{DCC}_{n_1+1}^-, \cdots, \text{DCC}_{n_2}^-}_{S_2}, \cdots, \\ &\underbrace{\text{DCC}_{n_{t-1}+1}^+, \cdots, \text{DCC}_{n_t}^+}_{S_t}, \cdots, \underbrace{\text{DCC}_{n_{M-1}+1}^- \cdots, \text{DCC}_N^-}_{S_M} \end{aligned} \quad (9)$$

Subsequently, a random permutation sequence of length N is generated according to (3), which is then partitioned into M segments. The length of the t-th segment is $|S_t|$. The DCCs in the t-th group S_t are permuted according to the t-th permutation sequence. Fig. 7 illustrates the permutation of DCCs in three groups S_k, S_l, S_m with the same sign.

Remark. *The proposed method of permuting consecutive DCCs with the same sign will not cause the quantized DC coefficients to fall outside the valid range (The proof is given in Appendix A).*

Moreover, the file size of a JPEG image will not be affected except for a slight variation resulting from byte alignment, as verified in our experiments.

3) Iterative grouping and swapping of DCCs: In addition to the permutation of DCCs introduced above, a technique for grouping and swapping DCCs iteratively is proposed to further distort JPEG images. The technique can provide flexible visual security controlled by the number of iterations. More iterations will lead to a more obscured JPEG image. The details of the j-th $(j = 1, 2, \cdots)$ iteration are described below.

Step 1. When $j-1$ iterations of grouping and swapping DCCs have been completed, all the DCCs are partitioned into $\lfloor N/(2j) \rfloor$ groups with $2j$ DCCs in each group.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2817065, IEEE Transactions on Multimedia

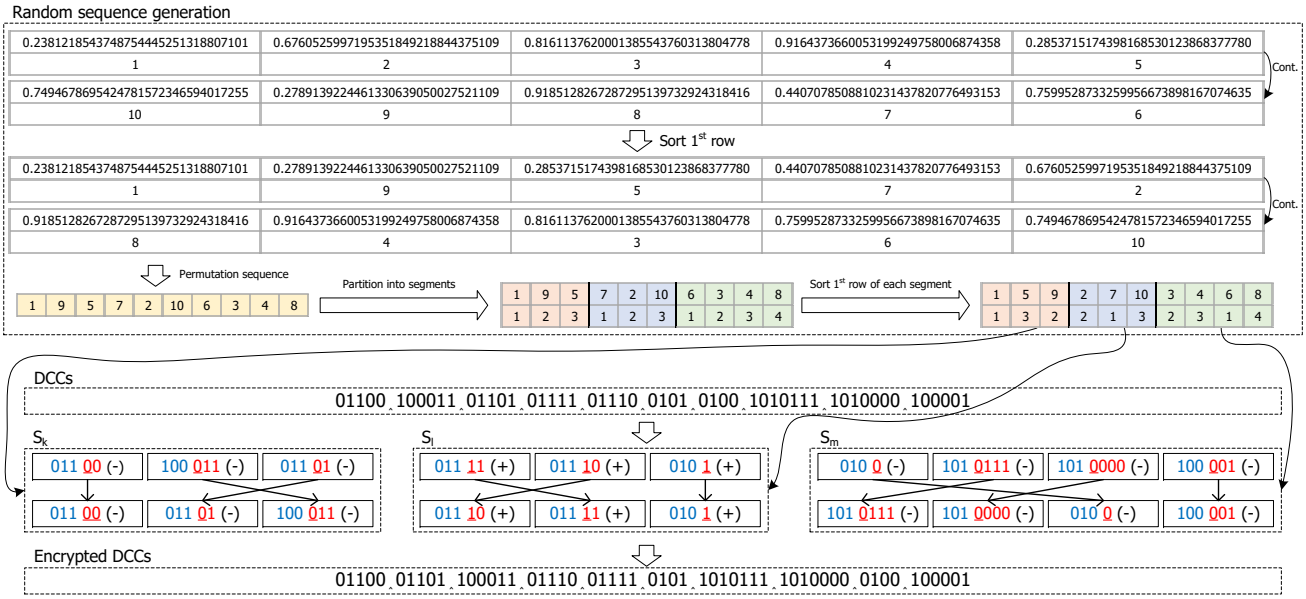


Fig. 7: Permutation of DCCs within three groups S_k, S_l, S_m with the same sign (blue bits represent DCHs while red bits represent DCAs, and $(*)$ denotes the sign of d_i, which indicated by the underlined red bit).

Step 2. Each of the DCC groups is evenly divided into two parts, i.e., the left half and the right half.

Step 3. Determine whether the left half and the right half can be exchanged without resulting in an overflow of quantized DC coefficients. If the left half and the right half are swapped, all the quantized DC coefficients in the group will still lie in the valid range; this DCC group is considered to be exchangeable.

Step 4. According to the random bitstream, which is determined by the parity of a permutation sequence, the left half and the right half of an exchangeable DCC group are swapped if the corresponding bit is '1'; otherwise, the group is kept intact if the bit '0' is received or it is nonexchangeable.

Step 5. When all the $\lfloor N/(2j) \rfloor$ DCC groups have been processed through Steps 2–4, if j is less than the predefined number of iterations, there follows the $(j+1)$-th iteration; otherwise, the iterative encryption of DCCs is completed.

The example shown in Fig. 8 demonstrates the grouping and swapping of DCCs. In the first iteration, six DCCs are divided into three groups. And the left half and the right half of each group will be swapped only when they are exchangeable and the corresponding bit in the chaotic bitstream is '1'.

As the number of iterations increases, the visual contents of a JPEG image are scrambled to a greater degree, and a more blurred image will be obtained. Of course, the computation time will be correspondingly increased. Therefore, the number of iterations should be chosen in accordance with the necessary visual security. In our experiments, we find that the JPEG image will appear as a random noise image when the number of iterations increases to 15. Similar to the previous procedure stated in Section III-D2, the file size of an encrypted JPEG image with iterative encryption will also not be largely altered except for byte alignment. In addition, it can be easily learned from the steps described above that the grouping and swapping of DCCs will not cause the quantized DC coefficients to fall outside the valid range.

E. Encryption of ACCs

For the encryption of ACCs, the ACCs within the same category, which is defined by the zero-run length, are globally scrambled across MCUs. Moreover, all MCUs excluding DCCs are randomly shuffled as a whole.

1) Permutation of ACCs within the same category: For an 8×8 DCT block, the zero-run lengths lie in the range $[0, 62]$. Thus, the ACCs can be classified into 63 categories according to their zero-run lengths. For each category of ACCs, a chaotic sequence of the same length as the number of ACCs inside it is generated according to (2) and then the corresponding permutation sequence obtained by (3) is employed to permute the ACCs within the corresponding category.

Fig. 9 is provided as an example to illustrate the permutation of ACCs within the same category from three different MCUs, $\text{MCU}_k, \text{MCU}_l, \text{MCU}_m$.

2) Permutation of MCUs excluding DCCs: Since the permutation of ACCs within the same category described in Section III-E1 does not alter the distribution of ACCs among all MCUs, the outline of a plaintext image can be easily obtained from the corresponding scrambled image using the methods proposed in [9], [12]. To prevent our proposed scheme from suffering the outline attacks, the order of the MCUs excluding DCCs is randomly rearranged.

As illustrated in Fig. 6, each MCU consists of one DCC and multiple ACCs. Permutation of the MCUs may result in an overflow of quantized DC coefficients because the DCCs will also be scrambled along with the ACCs. Therefore, the DCCs have to be excluded from each MCU before permutation. All the MCUs without DCCs are globally permuted as a whole

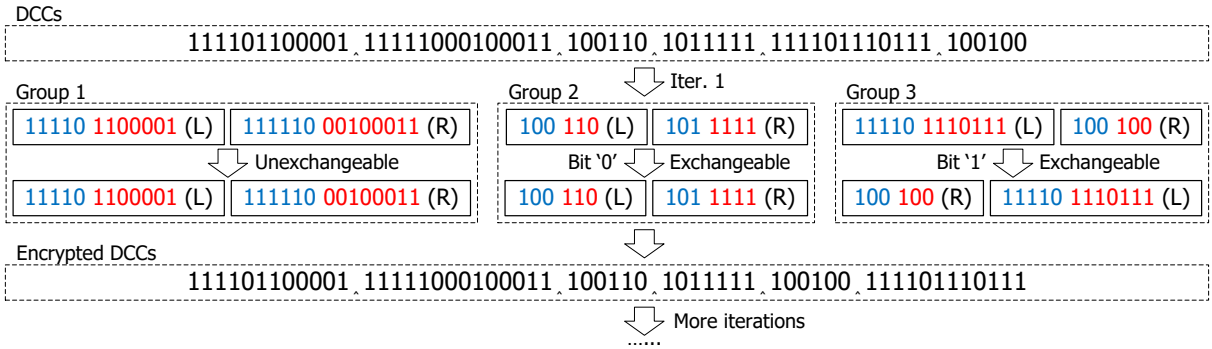


Fig. 8: Grouping and swapping of DCCs (blue bits represent DCHs while red bits represent DCAs, (L) and (R) denote the left half and right half of a DCC group respectively).

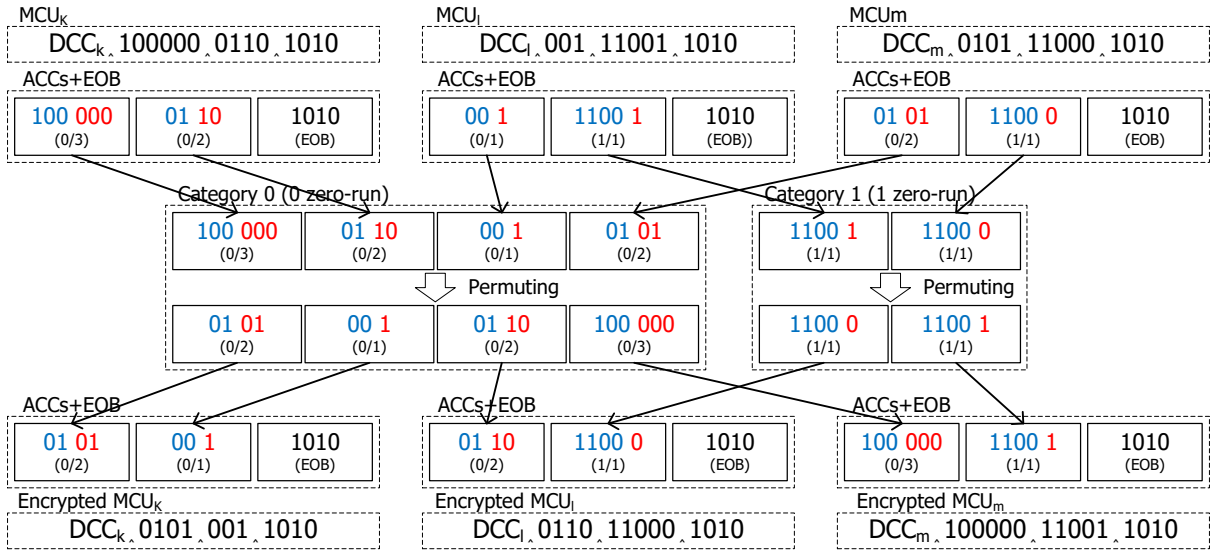


Fig. 9: Permutation of the ACCs within the same category from three different MCUs, i.e., $\mathrm{MCU}_k, \mathrm{MCU}_l, \mathrm{MCU}_m$ (x/y represents run/size, blue bits represent ACHs while red bits represent ACAs).

according to the random sequence generated by (2) and (3). Thus, the order of MCUs will be changed, while the relative positions of the ACCs within each MCU are kept intact.

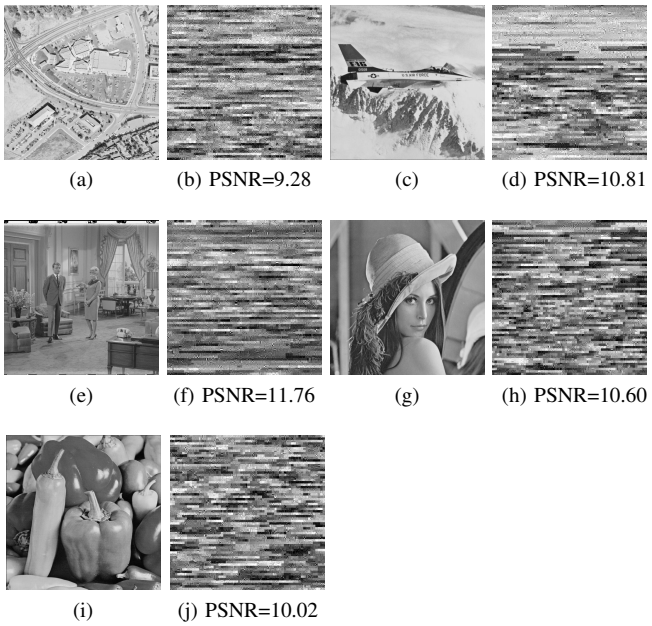
IV. Experimental Results and Analysis

In our experiments, five widely used images of size 512×512 (i.e., Aerial, Airplane, Couple, Lena and Pepper) downloaded from the USC-SIPI image database are employed to compare our proposed method with the related state-of-the-art works. Additionally, images from four image databases with great diversity, including BOWS-2, CorelDraw, UCID [28] and Ground Truth, are used to further demonstrate the effectiveness of our method. All images are converted into grayscale JPEG images using OpenCV with a quality factor of 85, except in the encryption of color JPEG images. The number of iterations used in the grouping and swapping of DCCs as described in Section III-D3 is set to 15 in most of our experiments, except in the flexible visual security experiments.

A. Visual encryption effect

The five JPEG images are encrypted using our proposed method with the parameters specified previously. As shown in Fig. 10, the visual contents of the plaintext JPEG images have been well blurred by the proposed encryption method. Without referring to the original images, it is difficult for us to distinguish one encrypted image from another. The PSNR (peak signal-to-noise ratio) values of the encrypted images are also presented in Fig. 10. The average PSNR values of all the encrypted images corresponding to the four image databases are provided in Table IV, from which we can observe that the PSNR values of the encrypted images are relatively low. Therefore, the images are greatly distorted by our proposed encryption method; hence, the visual security is guaranteed.

Moreover, as described in Section III-D3, the number of iterations of grouping and swapping of DCCs can be adjusted to obtain flexible visual security. As shown in Fig. 11, the visual contents of the encrypted JPEG images become more blurred as the number of iterations increases. It may be difficult to distinguish the encrypted images with 15 iterations shown in Fig. 11 (c), (g) from those with 32 iterations shown in Fig. 11 (d), (h). Thus, a compromise between the visual security and computational efforts should be made to meet the needs of practical applications.

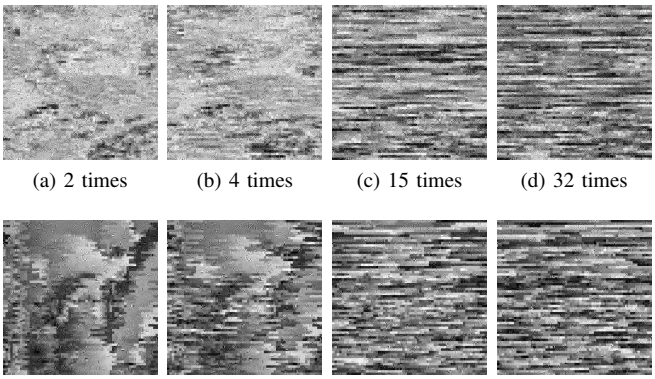


(a) (b) PSNR=9.28 (c) (d) PSNR=10.81

(e) (f) PSNR=11.76 (g) (h) PSNR=10.60

(i) (j) PSNR=10.02

Fig. 10: Five plaintext JPEG images and the corresponding encrypted images: (a) Aerial, (b) encrypted Aerial, (c) Airplane, (d) encrypted Airplane, (e) Couple, (f) encrypted Couple, (g) Lena, (h) encrypted Lena, (i) Pepper, and (j) encrypted Pepper.



(a) 2 times (b) 4 times (c) 15 times (d) 32 times

(e) 2 times (f) 4 times (g) 15 times (h) 32 times

Fig. 11: Encrypted Aerial (top) and Lena (bottom) with different number of iterations of grouping and swapping DCCs.

B. File format compatibility

According to the brief analysis presented in Section III-D1, the quantized DC coefficients have lower and upper bounds, both of which are related to the JPEG quality factor. The percentages of the quantized DC coefficients outside the valid range occurring in the JPEG images encrypted with different methods are given in Table V, where Ong+ and Ong− respectively indicate whether or not manipulations of the DC coefficients, AC coefficients and quantization table are employed to further distort a JPEG image (please refer to Section 3.5 in [21]).

As shown in Table V, almost all the quantized DC coefficients decoded from the encrypted JPEG images generated using Qian's [17], Ong+'s [21], and Cheng's [23] methods

TABLE IV: Average PSNRs and file size changes for four image databases ('+' means an increase in file size).

Indices	Image databases			
	BOWS-2	CorelDraw	UCID	Ground Truth
Resolution	512×512	768×512	512×384	756×504
Number of images	10000	917	885	557
PSNR	12.28	10.48	10.01	11.22
File size change (%)	+0.026	+0.018	+0.021	+0.014

TABLE V: Percentage of the quantized DC coefficients outside the valid range in encrypted JPEG images.

Encryption Methods	DC outside the valid range (%)				
	Aerial	Airplane	Couple	Lena	Pepper
Minemura [12]	0	0	0	0	0
Qian [17]	98.17	87.48	98.80	96.36	99.17
Ong− [21]	0	0	0	0	0
Ong+ [21]	99.73	99.27	99.90	99.98	100
Cheng [23]	98.51	97.41	94.17	96.61	95.65
Li [24]	5.81	11.74	1.07	2.42	4.39
Ours	0	0	0	0	0

fall outside the valid range. Additionally, a slight overflow of quantized DC coefficients occurs in Li's method [24]. The existence of the quantized DC coefficients outside the valid range may cause potential compatibility problems as demonstrated in Fig. 2. In this sense, our approach has improved compatibility. Although there is no overflow of quantized DC coefficients during decoding the encrypted JPEG images generated with Ong−'s [21] method, the number of AC coefficients accommodated in an 8×8 DCT block may exceed 63 after encryption, which may destroy the JPEG format compatibility. Moreover, the file size of the JPEG image encrypted with Ong−'s [21] method may be significantly altered, which will be discussed in Section IV-C.

C. File size preservation

As is well-known, JPEG is the most commonly used image format because it can achieve good compression with little perceptible loss in image quality. However, many existing encryption methods for JPEG images result in a noticeable increase in file size, which may be an undesirable consequence, particularly when large numbers of images are involved. The experimental results of the changes in the file sizes of JPEG images due to the encryption with different methods are shown in Table VI.

For the Ong+'s and Ong−'s [21] methods, extra pairs of (runlength/size, amplitude) are added to the bitstream of a JPEG image to record supplementary information necessary for reversible decryption, which will clearly increase its file size. Moreover, we can observe a significant increase in the file size of a JPEG image encrypted with Li's method [24]. The reason for this result is that both the category address mapping for the quantized DC coefficients and DCT block shuffling will destroy the statistical correlations of neighboring DC coefficients. However, a substantial increase in the file size is effectively prevented by the region-basis permutation proposed by Minemura [12].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2817065, IEEE Transactions on Multimedia

TABLE VI: Percentage of file size change resulting from JPEG encryption ('+' means an increase, and '−' means a decrease).

Encryption Methods	File size change (%)				
	Aerial	Airplane	Couple	Lena	Pepper
Minemura [12]	+0.088	-0.075	+0.128	+0.172	+0.085
Qian [17]	+0.025	-0.024	+0.026	+0.095	+0.015
Ong− [21]	+0.950	+1.694	-0.224	+1.339	+1.382
Ong+ [21]	+1.004	+1.729	-0.157	+1.330	+1.374
Cheng [23]	+0.055	-0.055	+0.107	+0.062	-0.008
Li [24]	+2.737	+6.424	+2.964	+3.811	+3.992
Ours	+0.034	-0.042	-0.006	+0.062	-0.033

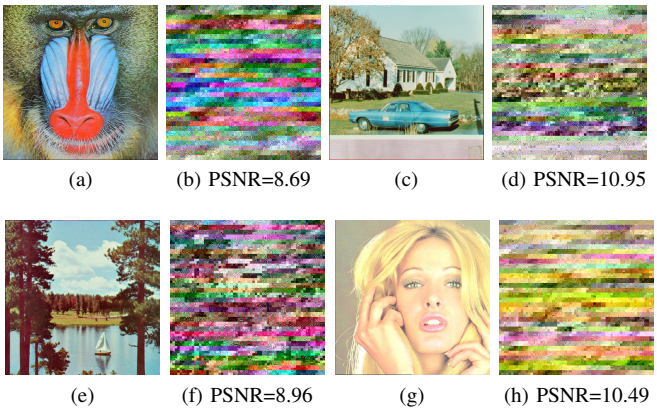
In theory, Qian's method [17], Cheng's method [23] and our proposed method will not change the file size. However, because of byte alignment defined in the JPEG standard [4], it can be observed from Table VI that there are still slight changes in the file size of the JPEG images encrypted with the above three methods. The average changes in the file size of the encrypted images corresponding to four image databases are presented in Table IV. This type of slight change may be inevitable as long as the bitstream of a JPEG image has been altered. Thus, it is not a flaw of the encryption methods; and these methods can be considered to have the feature of file size preservation.

D. Computational cost

Our proposed encryption method consists of four procedures. For permutation of consecutive DCCs with the same sign, both the traversal of N DCCs, which is needed in order to divide all DCCs into groups according to their signs, and the permutation of the DCCs within each group are linear time. For iterative grouping and swapping of DCCs, N DCCs are partitioned into $\lfloor N/(2j) \rfloor$ groups in the j-th iteration, then the left half and the right half of each group are swapped only when they are exchangeable and the corresponding chaotic bit is '1'. Each iteration has order of N time complexity, and 15 iterations are employed in our experiments. For permutation of ACCs within the same category, both the traversal of all ACCs ($\leq 63N$), which is needed in order to classify them into 63 categories, and the permutation of the ACCs within each category are linear time. Moreover, the permutation of MCUs excluding DCCs has linear time complexity. In conclusion, the time complexity of our method is $O(N)$.

E. Encryption of color JPEG images

Our proposed method can be easily extended to encrypt color JPEG images. When the proposed method is employed to encrypt a color JPEG image, the entropy encoded data corresponding to the illuminance component Y, the chrominance components Cb and Cr are processed respectively in the same way to encrypt a grayscale JPEG image. In our experiments, four 512×512 color images, i.e., Baboon, House, Sailboat and Tiffany, downloaded from the USC-SIPI image database are converted into color JPEG images using OpenCV with a quality factor of 85. The plaintext color images and the corresponding encrypted ones together with their PSNR values



(a) (b) PSNR=8.69 (c) (d) PSNR=10.95

(e) (f) PSNR=8.96 (g) (h) PSNR=10.49

Fig. 12: Four color plaintext JPEG images and the corresponding encrypted images: (a) Baboon, (b) encrypted Baboon, (c) House, (d) encrypted House, (e) Sailboat, (f) encrypted Sailboat, (g) Tiffany and (h) encrypted Tiffany.

TABLE VII: Percentage of file size change when our proposed encryption method is applied to color JPEG images.

Images	File size change (%)				
	50	60	70	80	85
Baboon	-0.006	+0.007	+0.031	+0.057	+0.022
House	+0.095	+0.054	+0.021	+0.031	-0.039
Sailboat	-0.023	-0.003	+0.074	+0.029	+0.013
Tiffany	-0.143	+0.048	+0.069	+0.032	+0.146

are shown in Fig. 12. And the percentages of the file size change are shown in Table VII.

As shown in Fig. 12, the visual contents of the plaintext images are well blurred after being encrypted with our method, and the PSNR values of the encrypted images are relatively low. Moreover, it can be observed from Table. VII that there is no noticeable change in the file sizes. Additionally, all the quantized DC coefficients will not fall outside the valid range as described in Section III-D2 and Section III-D3.

F. Comparison of Algorithm Features

To present an overview of the differences in features between the existing state-of-the-art JPEG encryption schemes and ours, the features of each scheme are summarized in Table VIII. The properties used for comparison include file size preservation, format compatibility, resisting the outline attacks, adaptive encryption, flexible visual security and the need for re-entropy encoding.

V. Security Analysis

The security of our proposed scheme is analyzed in this section. As with image encryption algorithms, there are two main kinds of attacks: brute-force attacks and cryptanalysis. A large key space provides the ability to defend against brute-force attacks. The statistical characteristics of a plaintext JPEG image are well confused and diffused through chaotic-based permutation of DCCs and ACCs, which helps our method to resist cryptanalysis.

TABLE VIII: Feature comparison of different methods.

Features	Encryption Methods						
	Minemura [12]	Qian [17]	Ong− [21]	Ong+ [21]	Cheng [23]	Li [24]	Ours
File size preservation	YES	YES	NO	NO	YES	NO	YES
Format compatibility	$\star\star\star$	$\star^{\dagger}$	$\star\star^{\dagger}$	$\star^{\ddagger}$	$\star^{\dagger}$	$\star\star^{\ddagger}$	$\star\star\star$
Resisting outline attack	YES	NO	YES	YES	YES	YES	YES
Adaptive encryption	NO	NO	NO	NO	NO	NO	YES
Flexible visual security	NO	NO	NO	NO	NO	NO	YES
No need for re-entropy encoding	NO	YES	NO	NO	YES	NO	YES

More $\star$ means better compatibility. Specifically, $\star^{\dagger}$ denotes that the encrypted JPEG images may not be successfully opened in popular image application software as illustrated in Fig. 4, and significant differences exist in the decoded images as shown in Fig. 2 because of a severe overflow of quantized DC coefficients; $\star^{\ddagger}$ means the number of AC coefficients in an 8×8 block may exceed 63 and significant differences exist in the decoded images; $\star\star^{\dagger}$ means only the number of AC coefficients in an 8×8 block may exceed 63; $\star\star^{\ddagger}$ indicates that only slight differences exist in the decoded images; $\star\star\star$ suggests none of the issues mentioned above exists in the decoded images.

A. Brute-force attack

A brute-force attack does not depend on a specific algorithm; rather, it depends only on the key space. On average, half of all the possible keys must be tried to achieve success. Our proposed method consists of adaptive encryption key generation based on chaotic logistic map and four scrambling operations on both the DCCs and the ACCs. Other chaotic sequence generation algorithms with larger key spaces can be easily incorporated into our adaptive encryption key generation method by making a slight modification; thus, the key space for the proposed scheme is determined by the possible permutations of each encryption operation, which are analyzed separately in the following.

1) Permutation of consecutive DCCs with the same sign: According to the description presented in Section III-D2, the DCCs with the same sign in each group S_t $(i = 1, 2, \cdots, M)$ are randomly permuted. Since there are a total of $|S_t|$ elements in group S_t, the total possible permutations P_1 of the DCCs can be calculated as

$$P_1 = \prod_{t=1}^{M} |S_t|! \tag{10}$$

2) Iterative grouping and swapping of DCCs: Let G denote the number of grouping and swapping iterations employed in the encryption of a JPEG image. As presented in Section III-D3, in the j-th $(j = 1, 2, \cdots, G)$ iteration, all DCCs are divided into $\lfloor N/(2j) \rfloor$ groups with $2j$ DCCs in each group. The left half and the right half of each group may be swapped depending on a chaotic bitstream and the possibility of exchange. If the percentage of the groups that can be exchanged is denoted by W_j in the j-th iteration, then the key space P_2 for grouping and swapping of DCCs is

$$P_2 = \prod_{j=1}^{G} 2^{\lfloor N/(2j) \rfloor \times W_j} \tag{11}$$

3) Permutation of ACCs within the same category: As stated in Section III-E1, all the ACCs are classified into 63 categories according to their zero-run lengths, and the ACCs in each category are internally scrambled. If the number of ACCs in each category is denoted as H_i $(i = 0, 1, \cdots, 62)$, then the possible permutations P_3 can be computed by

$$P_3 = \prod_{i=0}^{62} H_i! \tag{12}$$

4) Permutation of MCUs excluding DCCs: According to the JPEG bitstream parsing presented in Section III-A, there are N MCUs in a grayscale JPEG image. All MCUs excluding DCCs are randomly scrambled. Thus, the total number P_4 of the permutations of MCUs is

$$P_4 = N! \tag{13}$$

Combining (10)– (13), the full key space P of our proposed scheme can be calculated as

$$P = P_1 \times P_2 \times P_3 \times P_4 \tag{14}$$

Although it can be seen from (14) that the key space P is related to the JPEG image to be encrypted, the value of P is generally very large, which makes brute-force attacks impractical. As an example, the values of P_1, P_2, P_3, P_4 and P for five typical JPEG images of size 512×512 with a quality factor of 85 are given in Table IX. We can conclude from Table IX that it may be difficult to successfully recover the plaintext JPEG image using a brute-force attack.

B. Differential cryptanalysis

To resist against differential cryptanalysis, we have to ensure that a small difference in the input, including an initial key and a plaintext image as illustrated in Fig. 5, will result in a significant difference in the corresponding encrypted images. Both NPCR (number of pixels change rate) and UACI (unified average changing intensity) [1] are commonly used to evaluate an image encryption method's sensitivity to the input.

Let $c_1(i, j)$ and $c_2(i, j)$ denote the pixels at the coordinates (i, j) of two encrypted images C_1 and C_2 with only a small difference between two initial keys or two plaintext images; then, the NPCR is defined as

$$\text{NPCR} = \frac{\sum_{i,j} D(i, j)}{W \times H} \times 100\% \tag{15}$$

and the UACI is calculated by

$$\text{UACI} = \frac{1}{W \times H} \sum_{i,j} \frac{|c_1(i, j) - c_2(i, j)|}{255} \times 100\% \tag{16}$$

where W and H are the width and height of the encrypted images C_1 or C_2, $D(i, j) = 1$ if $c_1(i, j) \neq c_2(i, j)$, and $D(i, j) = 0$ if $c_1(i, j) = c_2(i, j)$.

Table X shows the NPCR and UACI results of five images encrypted by the proposed method, where 'NPCR (key)' and

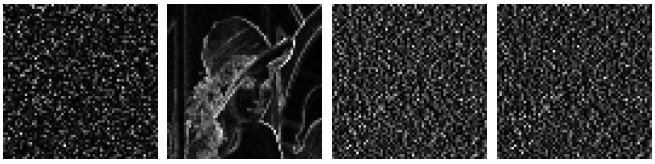
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2817065, IEEE Transactions on Multimedia

TABLE IX: The key space for five typical JPEG images.

JPEG images	Aerial	Airplane	Couple	Lena	Pepper
P_1	5.25×10^{782}	6.68×10^{1068}	7.55×10^{1009}	7.86×10^{1233}	6.06×10^{1363}
P_2	7.60×10^{1864}	1.40×10^{1884}	4.12×10^{1991}	5.13×10^{1785}	5.91×10^{1803}
P_3	2.10×10^{400945}	4.39×10^{223247}	8.17×10^{271216}	2.14×10^{221167}	5.43×10^{235349}
P_4	3.64×10^{13019}	3.64×10^{13019}	3.64×10^{13019}	3.64×10^{13019}	3.64×10^{13019}
P	3.05×10^{416612}	1.50×10^{239220}	9.24×10^{287237}	3.15×10^{237206}	7.08×10^{251536}

TABLE X: The values of NPCR and UACI.

JPEG images	Aerial	Airplane	Couple	Lena	Pepper
NPCR (key)	99.07%	99.23%	99.44%	99.53%	99.46%
UACI (key)	30.24%	23.42%	24.47%	26.22%	27.00%
NPCR (image)	98.98%	99.23%	99.44%	99.45%	99.48%
UACI (image)	30.05%	23.14%	25.75%	25.77%	27.03%



(a) Minemura [12] (b) Qian [17] (c) Ong− [21] (d) Ong+ [21]

(e) Cheng [23] (f) Li [24] (g) Ours

Fig. 13: Outline images generated using EAC attack.

'UACI (key)' correspond to the values when only a single digit of the initial key is changed, and 'NPCR (image)' and 'UACI (image)' are the values when only one pixel of the plaintext image is altered. It can be observed that even if only slight variations exist in the input, the values of NPCR are extremely close to their theoretical maximum values, which implies that the proposed method may generate almost completely different encrypted images; in addition, the values of UACI are also large, which implies that the average intensity differences between two encrypted images are significant. Thus, a high security level of the proposed method to withstand differential attacks can be ensured.

C. Known plaintext attack

If one secret key is used to encrypt multiple JPEG images, some existing encryption techniques may be insecure against known plaintext attacks. However, it may be desired that many JPEG images can be encrypted with the same key, especially in cloud-based applications. In our proposed scheme, the key applied to encrypt a JPEG image is constructed by combining the user-chosen initial key and the hash code of the image-content-related statistics, as described in Section III-C. Thus, it is secure from known plaintext attacks to use the proposed method to encrypt multiple images with only one initial key.

D. Shannon entropy analysis

As shown in Fig. 10 and Fig. 12, the visual contents of the JPEG images, whether they are grayscale or color, have been well confused with our bitstream-based encryption. The noise-like encrypted images do not reveal any information about the original images. The Shannon entropy of five encrypted 8-bit grayscale JPEG images, i.e., Aerial, Airplane, Couple, Lena and Pepper, are 7.63, 7.69, 7.69, 7.80 and 7.80. All the Shannon entropy is near 8, which is the theoretical maximum value. A high Shannon entropy value means a considerable randomness. Hence, from a statistical perspective, no original image information is leaked in the encrypted images.

E. Outline attack

Based on the fact that the number of non-zero AC coefficients in a DCT block correlates with some characteristics of the corresponding block, such as texture or edge information, a method called non-zero counting attack (NZCA) was proposed to obtain the outline image of an original JPEG image from the corresponding encrypted one [9]. Later, three novel methods, including non-zero coefficient count, energy of AC coefficients and position of last non-zero AC coefficient, were presented in [12] to improve the fidelity of the outline image produced by NZCA.

In our experiments, EAC is used to evaluate the security of the resistance to outline attacks because EAC always provides the best visual result. The outline images generated by EAC attack are illustrated in Fig. 13. As shown in Fig. 13, our proposed method and most of the related state-of-the-art methods except for the encryption method presented in [17] are safe against outline attacks. The reason is that the distribution of AC coefficients has been changed among DCT blocks or the positions of DCT blocks have been scrambled.

VI. Conclusion

JPEG is generally not only the default format of the images produced by many image capture devices but also the most commonly used image format in our daily lives. Image encryption techniques can be utilized to ensure the security and privacy of valuable JPEG images. In this paper, we present a novel bitstream-based JPEG image encryption method. The experimental results and security analysis demonstrated the improved performance of our scheme. The main contributions of this paper are as follows:

1) Novel encryption techniques toward DC and AC codes are proposed, which contribute to the new JPEG image encryption scheme presented in this paper. The file size of an encrypted JPEG image generated with the proposed method

can be preserved except for slight variations because of byte alignment. Moreover, better format compatibility with the JPEG standard is obtained.

2) All the encryption operations are conducted directly on the bitstream without the need of entropy encoding once again to generate an encrypted JPEG image. Additionally, the number of iterations of grouping and swapping DC codes can be adjusted to reach a compromise between visual security and computational efforts.
3) An image-content-related encryption key is employed to provide better security against known plaintext attacks. The proposed scheme can provide a large key space enough to defend against brute-force attacks. And the chaotic-based permutations help our method to resist cryptanalysis.

APPENDIX
PROOF OF REMARK IN SECTION III-D2

Mathematical induction on the number of DCC groups with the same sign undergoing permutation is used.

i) True for 1 (base case):

Let d_i^1 $(i = 1, 2, \cdots, N)$ denote the i-th quantized DC difference after only the DCCs in the first group S_1 are randomly scrambled. Without loss of generality, all the DCCs in S_1 are assumed to be non-negative, i.e., $d_i \geq 0$ $(i = 1, 2, \cdots, n_1)$. The permutation of S_1 will not alter its values, that is, $\{d_1^1, d_2^1, \cdots, d_{n_1}^1\} = \{d_1, d_2, \cdots, d_{n_1}\}$. Hence,

$$\begin{aligned} \mathrm{QD}_i^1 &= \sum\nolimits_{j=1}^{i} d_j^1 \geq 0 \geq \mathrm{MIN}; \\ \mathrm{QD}_i^1 &= \sum\nolimits_{j=1}^{i} d_j^1 \leq \sum\nolimits_{j=1}^{n_1} d_j^1 = \sum\nolimits_{j=1}^{n_1} d_j \\ &= \mathrm{QD}_{n_1} \leq \mathrm{MAX} \end{aligned} \tag{17}$$

where $i = 1, 2, \cdots, n_1$ and QD_i^1 denotes the encrypted quantized DC coefficient after S_1 is scrambled.

Moreover, because the DCCs except for the first n_1 ones are not modified during the permutation of S_1, $d_j^1 = d_j$ $(j = n_1 + 1, n_1 + 2, \cdots, N)$. Then, we obtain

$$\begin{aligned} \mathrm{QD}_i^1 &= \sum_{j=1}^{i} d_j^1 = \sum_{j=1}^{n_1} d_j^1 + \sum_{j=n_1+1}^{i} d_j^1 \\ &= \sum_{j=1}^{n_1} d_j + \sum_{j=n_1+1}^{i} d_j = \sum_{j=1}^{i} d_j = \mathrm{QD}_i \end{aligned} \tag{18}$$

where $i = n_1 + 1, n_1 + 2, \cdots, N$.

It can be concluded from (8), (17) and (18) that all the encrypted quantized DC coefficients QD_i^1 $(i = 1, 2, \cdots, N)$ will fall in the valid range after the permutation of S_1.

ii) True for t if true for $t-1$ (inductive step):

Let d_i^{t-1} denote the i-th quantized DC difference after the DCCs in the first $t-1$ $(t \geq 2)$ groups are internally permuted. Suppose all the encrypted quantized DC coefficients QD_i^{t-1} lie in the valid range after S_1–S_{t-1} are scrambled, say,

$$\mathrm{MIN} \leq \mathrm{QD}_i^{t-1} = \sum\nolimits_{j=1}^{i} d_j^{t-1} \leq \mathrm{MAX} \tag{19}$$

where $i = 1, 2, \cdots, N$.

Let d_i^t denote the i-th quantized DC difference, and QD_i^t represent the corresponding quantized DC coefficients after the DCCs in the extra group S_t are also internally scrambled, where $i = 1, 2, \cdots, N$. Because the permutation of S_t is carried out within S_t, the first n_{t-1} values will not change, i.e., $d_j^t = d_j^{t-1}$ $(j = 1, 2, \cdots, n_{t-1})$ Thus,

$$\mathrm{MIN} \leq \mathrm{QD}_i^t = \sum_{j=1}^{i} d_j^t = \sum_{j=1}^{i} d_j^{t-1} = \mathrm{QD}_i^{t-1} \leq \mathrm{MAX} \tag{20}$$

where $i = 1, 2, \cdots, n_{t-1}$.

Without loss of generality, all the DCCs in S_t are assumed to be non-negative, i.e., $d_i^{t-1} \geq 0$ $(i = n_{t-1} + 1, n_{t-1} + 2, \cdots, n_t)$. The permutation of S_t will not alter its values, in other words, $\{d_{n_{t-1}+1}^t, d_{n_{t-1}+2}^t, \cdots, d_{n_t}^t\} = \{d_{n_{t-1}+1}^{t-1}, d_{n_{t-1}+2}^{t-1}, \cdots, d_{n_t}^{t-1}\}$. Hence,

$$\begin{aligned} \mathrm{QD}_i^t &= \sum_{j=1}^{i} d_j^t = \sum_{j=1}^{n_{t-1}} d_j^t + \sum_{j=n_{t-1}+1}^{i} d_j^t \\ &\geq \sum_{j=1}^{n_{t-1}} d_j^t = \mathrm{QD}_{n_{t-1}}^t = \mathrm{QD}_{n_{t-1}}^{t-1} \geq \mathrm{MIN}; \\ \mathrm{QD}_i^t &= \sum_{j=1}^{i} d_j^t = \sum_{j=1}^{n_{t-1}} d_j^t + \sum_{j=n_{t-1}+1}^{i} d_j^t \\ &= \sum_{j=1}^{n_{t-1}} d_j^{t-1} + \sum_{j=n_{t-1}+1}^{i} d_j^t \leq \sum_{j=1}^{n_{t-1}} d_j^{t-1} + \sum_{j=n_{t-1}+1}^{n_t} d_j^t \\ &= \sum_{j=1}^{n_{t-1}} d_j^{t-1} + \sum_{j=n_{t-1}+1}^{n_t} d_j^{t-1} = \mathrm{QD}_{n_t}^{t-1} \leq \mathrm{MAX} \end{aligned} \tag{21}$$

where $i = n_{t-1} + 1, n_{t-1} + 2, \cdots, n_t$.

Moreover, because the DCCs except for the first n_t ones are not modified during the permutations of S_1–S_t, $d_j^t = d_j^{t-1}$ $(j = n_t + 1, n_t + 2, \cdots, N)$. Then, we obtain

$$\begin{aligned} \mathrm{QD}_i^t &= \sum_{j=1}^{i} d_j^t = \sum_{j=1}^{n_{t-1}} d_j^t + \sum_{j=n_{t-1}+1}^{n_t} d_j^t + \sum_{j=n_t+1}^{i} d_j^t \\ &= \sum_{j=1}^{n_{t-1}} d_j^{t-1} + \sum_{j=n_{t-1}+1}^{n_t} d_j^{t-1} + \sum_{j=n_t+1}^{i} d_j^{t-1} \\ &= \sum_{j=1}^{i} d_j^{t-1} = \mathrm{QD}_i^{t-1} \end{aligned} \tag{22}$$

where $i = n_t + 1, n_t + 2, \cdots, N$.

It can be concluded from (20)–(22) that all the encrypted quantized DC coefficients QD_i^t $(i = 1, 2, \cdots, N)$ after the permutations of S_1–S_t will still lie in the valid range.

□

REFERENCES

[1] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons & Fractals*, vol. 21, no. 3, pp. 749–761, 2004.
[2] Y. Zhou, L. Bao, and C. P. Chen, "A new 1D chaotic system for image encryption," *Signal Processing*, vol. 97, pp. 172–182, 2014.
[3] W. Zhang, H. Wang, D. Hou, and N. Yu, "Reversible data hiding in encrypted images by reversible image transformation," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1469–1479, 2016.

[4] *Digital Compression and Coding of Continuous-tone Still Images Requirements and Guidelines*, International Telecommunication Union Std. CCITT Recommendation T.81, 1992.
[5] L. Tang, "Methods for encrypting and decrypting MPEG video data efficiently," in *Proceedings of the 4th ACM International Conference on Multimedia (MULTIMEDIA'96)*, 1996, pp. 219–229.
[6] S. Lian, J. Sun, and Z. Wang, "A novel image encryption scheme based-on JPEG encoding," in *Proceedings of the 8th International Conference on Information Visualisation (IV'04)*, 2004, pp. 217–220.
[7] J. M. Rodrigues, W. Puech, and A. G. Bors, "Selective encryption of human skin in JPEG images," in *Proceedings of the 13th IEEE International Conference on Image Processing (ICIP'06)*, 2006, pp. 1981–1984.
[8] F. Dufaux and T. Ebrahimi, "Toward a secure JPEG," in *Proceedings of SPIE, Applications of Digital Image Processing XXIX*, vol. 6312, 2006, pp. 63 120K–1–8.
[9] W. Li and Y. Yuan, "A leak and its remedy in JPEG image encryption," *International Journal of Computer Mathematics - Computer Vision and Pattern Recognition*, vol. 84, no. 9, pp. 1367–1378, 2007.
[10] X. Niu, C. Zhou, J. Ding, and B. Yang, "JPEG encryption with file size preservation," in *Proceedings of the 4th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP'08)*, 2008, pp. 308–311.
[11] B. K. Shreyamsha Kumar and C. R. Patil, "JPEG image encryption using fuzzy PN sequences," *Signal, Image and Video Processing*, vol. 4, no. 4, pp. 419–427, 2010.
[12] K. Minemura, Z. Moayed, K. Wong, X. Qi, and K. Tanaka, "JPEG image scrambling without expansion in bitstream size," in *Proceedings of the 19th IEEE International Conference on Image Processing (ICIP'12)*, 2012, pp. 261–264.
[13] A. Unterweger and A. Uhl, "Length-preserving bit-stream-based JPEG encryption," in *Proceedings of the 14th ACM Multimedia and Security Workshop (MMSec'12)*, 2012, pp. 85–90.
[14] S. Y. Ong, K. Minemura, and K. S. Wong, "Progressive quality degradation in JPEG compressed image using DC block orientation with rewritable data embedding functionality," in *Proceedings fo 20th the IEEE International Conference on Image Processing (ICIP'13)*, 2013, pp. 4574–4578.
[15] S. Auer, A. Bliem, D. Engel, A. Uhl, and A. Unterweger, "Bitstream-based JPEG encryption in real-time," *International Journal of Digital Crime and Forensics*, vol. 5, no. 3, pp. 1–14, 2013.
[16] D. Zhang and F. Zhang, "Chaotic encryption and decryption of JPEG image," *Optik - International Journal for Light and Electron Optics*, vol. 125, no. 2, pp. 717–720, 2014.
[17] Z. Qian, X. Zhang, and S. Wang, "Reversible data hiding in encrypted JPEG bitstream," *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1486–1491, 2014.
[18] B. Kishore, B. K. Shreyamsha Kumar, and C. R. Patil, "FPGA based simple and fast JPEG encryptor," *Journal of Real-Time Image Processing*, vol. 10, no. 3, pp. 551–559, 2015.
[19] M. Ghadi, L. Laouamer, and T. Moulahi, "Enhancing digital image integrity by exploiting JPEG bitstream attributes," *Journal of Innovation in Digital Ecosystems*, vol. 2, no. 12, pp. 20–31, 2015.
[20] H. Cheng, X. Zhang, J. Yu, and F. Li, "Markov process based retrieval for encrypted JPEG images," in *Proceedings of the 10th International Conference on Availability, Reliability and Security (ARES'15)*, 2015, pp. 417–421.
[21] S. Ong, K. Wong, X. Qi, and K. Tanaka, "Beyond format-compliant encryption for JPEG image," *Signal Processing: Image Communication*, vol. 31, pp. 47–60, 2015.
[22] L. Yuan, P. Korshunov, and T. Ebrahimi, "Privacy-preserving photo sharing based on a secure JPEG," in *Proceedings of the 3rd International Workshop on Security and Privacy in Big Data (BigSecurity'15)*, 2015, pp. 185–190.
[23] H. Cheng, X. Zhang, J. Yu, and Y. Zhang, "Encrypted JPEG image retrieval using block-wise feature comparison," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 111–117, 2016.
[24] S. Li and Y. Zhang, "Quantized DCT coefficient category address encryption for JPEG image," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 4, pp. 1790–1806, 2016.
[25] S. Çiftçi, A. O. Akyüz, and T. Ebrahimi, "A reliable and reversible image privacy protection based on false colors," *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 68–81, 2018.
[26] B. Bhargava, C. Shi, and S.-Y. Wang, "MPEG video encryption algorithms," *Multimedia Tools and Applications*, vol. 24, no. 1, pp. 57–79, 2004.
[27] T. Granlund and the GMP development team, "Gnu multiple precision arithmetic library," 2016. [Online]. Available: https://gmplib.org
[28] G. Schaefer and M. Stich, "UCID: an uncompressed color image database," in *Proceedings of SPIE, Storage and Retrieval Methods and Applications for Multimedia 2004*, vol. 5307, 2003, pp. 472–480.

Junhui He received his B.S., M.S., and Ph.D. degrees from Central South University, South China University of Technology, and Sun Yat-Sen University, China, in 1997, 2000, and 2006, respectively. He is currently an associate professor with the School of Computer Science and Engineering, South China University of Technology. His current research interests include multimedia security, steganography and steganalysis.

Shuhao Huang received his B.S. degree in information security from Jinan University, China, in 2015. He is currently pursuing his M.S. degree in computer science at South China University of Technology. His current research interests include multimedia security and applications.

Shaohua Tang (M'99) received the B.Sc. and M.Sc. degrees in applied mathematics from the South China University of Technology, China, in 1991 and 1994, respectively, and the Ph.D. degree in communication and information system from the South China University of Technology, in 1998. He was a Visiting Scholar with North Carolina State University, USA, and a Visiting Professor with the University of Cincinnati, USA. He has been a Full Professor with the School of Computer Science and Engineering, South China University of Technology, since 2004. He has authored or co-authored over 100 technical papers in journals and conference proceedings. His current research interests include information security, data security, and privacy preserving in cloud computing and big data. He is a member of the IEEE Computer Society.

Jiwu Huang (M'98-SM'00-F'16) received the B.S. degree from Xidian University, Xian, China, in 1982, the M.S. degree from Tsinghua University, Beijing, China, in 1987, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1998. He was with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China. He is currently a Professor with the College of Information Engineering, Shenzhen University, Shenzhen, China. His current research interests include multimedia forensics and security. He is also a member of the IEEE Circuits and Systems Society Multimedia Systems and Applications Technical Committee and the IEEE Signal Processing Society Information Forensics and Security Technical Committee. He served as an Associate Editor of the IEEE Transactions on Information Forensics and Security from 2010 to 2014. He was a General Co-Chair of the IEEE Workshop on Information Forensics and Security in 2013. He is a Fellow of IEEE.