applied sciences

MDPI

Article

Content and Privacy Protection in JPEG Images by Reversible Visual Transformation

Xin Cao [1], Yuxuan Huang [1], Hao-Tian Wu [1,*] and Yiu-ming Cheung [2]

[1] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China; mscaoxin@mail.scut.edu.cn (X.C.); 201820133045@mail.scut.edu.cn (Y.H.)

[2] Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong, China; ymc@comp.hkbu.edu.hk

* Correspondence: wuht@scut.edu.cn; Tel.: +86-18320739339

Received: 17 August 2020; Accepted: 24 September 2020; Published: 27 September 2020

Abstract: With the popularity of cloud computing and social networks, more and more JPEG images are stored and distributed. Consequently, how to protect privacy and content in JPEG images has become an important issue. Although traditional encryption schemes can be employed, the file format of JPEG images is changed so that their usage may be affected. In this paper, a reversible visual transformation algorithm is proposed to protect content in JPEG images. Specifically, the DC coefficient in each user-selected block is modified, while the information required to recover it is reversibly hidden into AC coefficients. Then the signs of AC coefficients in the selected blocks are flipped and the blocks are further scrambled with a secret key. By embedding the location information of the selected blocks in a transformed image, the original image can be exactly recovered when needed. Besides, regions to be protected can be arbitrarily chosen without substantially affecting the rest of the image. The experimental results on a set of JPEG images validate the efficacy and reversibility of the proposed algorithm. In addition, good performance is achieved in terms of invisibility of the protected content, image quality, file size preservation and security.

Keywords: privacy protection; social network; reversible data hiding; visual encryption; block scrambling

1. Introduction

With the popularity of cloud computing and social networks, more and more images are stored and distributed on the web. There are quite a few image formats, among which a popular one is JPEG [1] for its high efficiency in compression. Compared with lossless formats such as bitmap (BMP), JPEG compression can significantly reduce the file size to save bandwidth without introducing noticeable degradation. With the dissemination of JPEG images, how to protect content and privacy for them has become an important issue. For instance, a user may want to share some JPEG images containing private information but directly exposing them in social networks may impair privacy.

In the literature, a number of methods (e.g., [2–13]) have been proposed for privacy protection in JPEG images, including selective encryption (e.g., [2]), image blurring, image pixelation, image masking, image morphing (e.g., [3]) and image warping (e.g., [4]). To keep the sensitive content from being exposed, these methods may change image content permanently and even degrade image quality. By contrast, it is advantageous for authorized receivers to recover the original images when needed. Hence, data hiding has been applied in JPEG image encryption to achieve such reversibility. For instance, scrambling and image data hiding are combined in Ong's algorithm [5]. Moreover, image masking and data hiding are combined in [6,7]. Nevertheless, the original image cannot be completely recovered with the algorithms proposed in [5–7]. Then, a privacy protection algorithm based on

scrambling the quantized Discrete Cosine Transform (DCT) coefficients was proposed in [8], but the JPEG file size is significantly increased. Reversible data hiding (RDH) (e.g., [14]) is applied in Joshi's algorithm [9] and Niimi's algorithm [10] so that the original JPEG images can be completely recovered. In particular, Joshi's algorithm uses a prediction-based RDH technique to achieve reversibility and protects sensitive regions by modifying pixel values in them. However, the size of the privacy-sensitive region (PSR) is restricted. Additionally, a large PSR may cause distortions of a non-sensitive region while a small PSR may increase the file size. In [10], Coltuc's reversible watermarking algorithm [14] is adopted to modify AC coefficients for content protection. Subsequently, two successive AC coefficients are further modified to mark the protected blocks. Although the performance of [10] for content protection was evaluated, the security has not been properly analyzed. As a matter of fact, the protected blocks can be easily identified and the AC coefficients may be restored by a brute force search.

To enhance the security, we improve the algorithm in [10] in two aspects. Firstly, the location information of protected blocks is generated instead of modifying two successive AC coefficients. By embedding the location information into the whole JPEG image with the RDH algorithm in [15], the protected blocks can be identified only after data extraction. Secondly, the protected blocks are scrambled with a secret key in our proposed algorithm so that it is more difficult for an adversary to recover the original image. Similar to the algorithm in [10], the information required for image recovery is reversibly hidden in the AC coefficients by adopting the algorithm in [16]. It turns out that the original images can only be obtained after extracting the hidden data with the secret key.

To evaluate the performance, invisibility of the protected content, image quality, reversibility and JPEG file size increment are measured. The experimental results show that the proposed algorithm provides good security against attacks. The protected content is invisible while the quality of the other blocks is well preserved. Moreover, the proposed algorithm outperforms the existing techniques in terms of reversibility and file size preservation. In the next section, some related work will be reviewed. Section 3 introduces details of the proposed algorithm. Section 4 presents the experiments and evaluation results. Finally, we draw conclusions in Section 5.

2. Related Work

The algorithms for privacy protection in JPEG images can be classified into two categories: (1) algorithms performed on bit stream (e.g., [2]), and (2) algorithms performed on the quantified DCT coefficients (e.g., [5–10]). Since modifications of the code stream may change the format of compressed images, the decryption should be completed before decoding and a dedicated decoder needs to be equipped. By contrast, algorithms processed on the quantified DCT coefficients are more flexible and protected images can be decoded directly.

Advanced encryption standard (AES) [17] is a classical and popular symmetric encryption algorithm, which was applied in a selective encryption of JPEG2000 in [2]. In the selective encryption algorithm, the bit stream is divided into different packets, and the most significant part of the entire packet data is encrypted by AES. This algorithm works effectively on JPEG2000, but it cannot be applied to JPEG images because of the different coding schemes. Nevertheless, AES can be applied in the encryption of JPEG images in another manner. The encryption is achieved by simply recognizing the JPEG code stream as common plaintext without focusing on the characteristics of JPEG. A series of operations such as byte substitution, row displacement, column mixing, round key addition and exclusive or are conducted on the plaintext. However, the visibility and the file size of JPEG images will be affected. Besides, it requires more time to encrypt an image with AES because of the complicated calculation process.

Alternatively, quite a few algorithms have been proposed by utilizing the characteristics of the JPEG coding scheme to achieve privacy protection. These algorithms are processed on the quantified DCT coefficients, and the proposed algorithm also belongs to this category. Specifically, the file size increment should be controlled within an acceptable range while achieving content protection and security against attacks.

The P3 algorithm [11] leverages sparsity and the high quality of JPEG images to divide a JPEG image into a public part and a secret part. The public part is exposed to the photo sharing service providers, while the secret part is encrypted and shared between the sender and the recipients. As a result, the two parts need to be stored separately, which introduces an extra cloud storage request for the secret part and complicates the file management system. In addition, the P3 algorithm only works on the whole-image level and it is inconvenient for users to select the protected regions.

Yuan's algorithm [12] allows users to select several regions of arbitrary shapes to be protected. The signs of the DCT coefficients in selected regions are modified to scramble the image and the auxiliary information for recovery is inserted into one or more application markers in the JPEG file header. Thus the privacy and its reversibility can be achieved. The data hiding approach is employed to avoid extra overhead to storage, but it causes a larger file increments compared with other algorithms.

Both Li's algorithm [7] and the algorithm based on false colors [18] modify the value of DCT coefficients to protect the sensitive information and record the difference for recovering. In Li's algorithm, a cartoon mask is used to cover a human face by calculating the DCT coefficients of the mask and replacing the original ones. The original value of the DCT coefficients and the difference are embedded in the JPEG image. In the process, the sparse representation is used to find a more concise expression for each original image signal, so the cost of file storage can be decreased. Nevertheless, F5 steganography [19] or another JPEG steganography algorithm (e.g., [20]) is applied in the embedding operation, but the original JPEG images cannot be recovered completely. Similarly, a scalable scrambling algorithm operating in the DCT domain is proposed in [13] within the JPEG codec. The goal is to ensure that people are no more identifiable while keeping their actions still understandable regardless of the image size.

The algorithm proposed in [18] substitutes false color for each original color value through a specific mapping relationship. As a result, detailed information, such as a human face, cannot be identified. Since the algorithm exploits the coherence of the false color version of the encrypted JPEG images and the original ones, the size of the protected content is reduced. Moreover, lossless image recovery is achieved since the difference and other auxiliary information is hidden in the JPEG image to be protected. However, the embedding operation may lead to a large cost of file storage. Since the privacy protection based on false colors is mainly used in video surveillance, the image outline is visible and some privacy information is exposed.

As the existing algorithms cannot fully meet the requirements of privacy protection in JPEG images, especially in reversibility and file size preservation, a reversible and effective algorithm is proposed in the following section, which achieves not only reversibility but also good performance in terms of invisibility of the protected content, image quality, file size preservation and security.

3. Proposed Algorithm for Content Protection in JPEG Images

The algorithm to be presented can be used to protect both grayscale and color JPEG images. The algorithm for grayscale images will be introduced only, which can be directly applied to the luminance component of a color image. The proposed algorithm consists of two parts, i.e., converting an original JPEG image into a visually transformed one and recovering the original JPEG image. In the following section, the two parts will be introduced in detail.

3.1. Generating the Visually Transformed JPEG Images

The flowchart of generating the visually transformed JPEG image from an original one is shown in Figure 1, in which user interactions are needed to select the regions to be protected. According to the user selection, the image blocks containing the selected regions can be identified and their positions can be recorded as binary location information. Then DC coefficients in the selected blocks are modified while the difference between the original and modified DC coefficients is reversibly embedded into the AC coefficients. After that, the signs of the AC coefficients in the selected blocks are flipped with a sequence of 1 s and −1 s generated with a secret key. The selected blocks are further scrambled but

the DC coefficients in them are kept unmoved. Finally, the protected JPEG image is generated by reversibly embedding the location information into it.

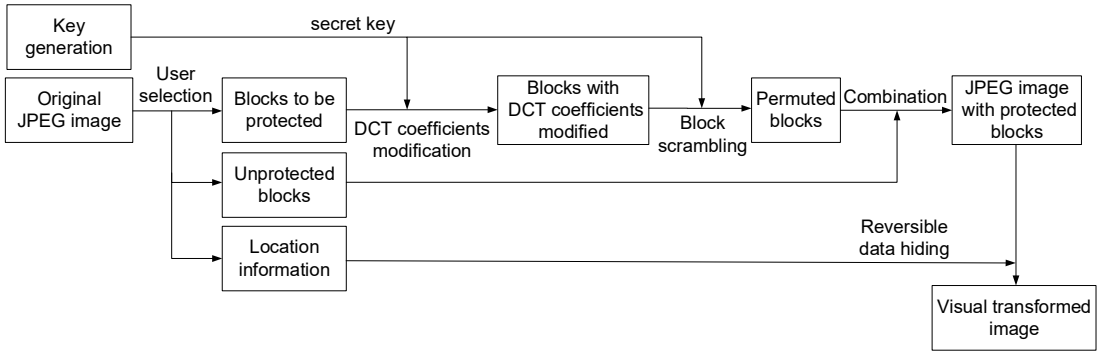

Figure 1. A flowchart of visual transformation for content and privacy protection in JPEG images.

3.1.1. Key Generation

For security concern, a secret key is used in the proposed visual transformation algorithm. The secret key is shared between the sender and receiver, or can be transmitted by using public-key cryptography. To generate different keys for different images to be protected, the secret key can be concatenated to the mega information of an individual image to generate a hash value of 128 bits by MD5 [21]. Then the hash value is used as the seed of a pseudo-random number generator (PRG) to generate random sequences in the flowing steps. In this way, the user can use the same key to obtain different seeds for different JPEG images to resist brute force attack [22].

3.1.2. User Interaction to Select the Regions to Be Protected

A JPEG image is divided into 8×8 blocks and each block is assigned with an index ranging from 0 to $n-1$, as there are n blocks in the JPEG image. When a user selects one or more rectangular regions to be protected, the location information of the selected regions will be recorded. For each selected region, the index of top left block, and the height and width in blocks are recorded. Suppose there are w blocks per row and h blocks per column in a JPEG image with n blocks. The location information of each block consists of three parts with the lengths as follows:

$$l = \lceil log_2\, n \rceil \tag{1}$$

$$l_1 = \lceil log_2\, w \rceil \tag{2}$$

$$l_2 = \lceil log_2\, h \rceil \tag{3}$$

where l, l_1, l_2 represent the lengths in bits for block index, region width and height, respectively. The recorded information of all selected regions is combined together to form a location map. Note that the length of the generated location map is calculated, which is denoted by *len* and embedded into the JPEG image prior to the location map.

3.1.3. Modifying the DC Coefficients

Differential pulse code modulation (DPCM) and Huffman coding are used to encode DC coefficients by exploiting the correlations between adjacent DC coefficients. To enhance the security while reducing the JPEG file size, all DC coefficients in selected regions are modified to similar values. For a DC coefficient d, the modification is conducted by

$$d' = d - delta', \tag{4}$$

where d' represents the modified DC coefficient and $delta'$ is obtained as follows. At first, the difference between d and the average (denoted by avg) of all DC coefficients in the selected blocks is calculated by

$$delta = d - avg, \tag{5}$$

where $delta$ is the original difference between d and avg. To reduce the data for representation, $delta$ is quantized to $delta'$ with the method mentioned in [23] by

$$delta' = \begin{cases} 8 \times round\left(\frac{delta}{8}\right) & , \ delta > 0 \\ 8 \times floor\left(\frac{delta}{8}\right) + 4, & delta < 0 \end{cases}, \tag{6}$$

where $delta'$ is the multiple of 4 so that it can be represented with fewer bits. For recovery, the value of $delta'$ is embedded into the AC coefficients in the same block by adopting an RDH algorithm proposed in [16]. Specifically, the binary value of $\frac{|delta'|}{4}$ is hidden in the AC coefficients and we can recover $delta'$ from $\frac{|delta'|}{4}$ by multiplying it with 4. In addition, the sign of $delta$ can be inferred from whether $\frac{|delta'|}{4}$ is odd or even. As the experimental results conducted in [24] show, hiding data in low-frequency coefficients helps to achieve a smaller file size, and the 3rd to 11th AC coefficients of each block are chosen to carry the bit values. Since each DCT coefficient ranges from −1023 to 1023, $delta$ is within [−2046, 2046], which indicates that $\frac{|delta'|}{4}$ can be expressed with 9 bits. For each AC coefficient a_j (j = 3, 4, 5, . . . , 11) and a bit value s (0 or 1), data embedding is performed by

$$a_j' = a_j * 2 + s. \tag{7}$$

Now we take the example as shown in Figure 2 to explain the aforementioned process. Suppose there are four blocks in the selected region and the DC coefficients are 37, 3, 20, 20 with an average value of 20. For the first DC coefficient, the value of $delta$ is 17 and the value of $delta$ is modified to 16 according to Equation (6). After obtaining the quantified $delta$, the DC coefficient is modified to 21 by calculating the difference between the original one and $delta'$. Finally, $\frac{|delta'|}{4}$ is represented in nine bits, which are embedded into nine AC coefficients of the same block with Equation (7). The cases of modifying the other DC coefficients are also shown in Figure 2.

3.1.4. Permutation of the Selected Blocks

A random sequence R1 of 1 s and −1 s is generated with a PRG, which takes a secret key as the seed. Then the signs of AC coefficients in the selected regions are flipped by multiplying the sequence of AC coefficients with R1, while the selected blocks are further scrambled with another random sequence, R2, generated with the same secret key. Specifically, the DC coefficients are kept unmoved and the lengths of the two random sequences depend on the number of selected blocks. Suppose there are n blocks in the selected region, the length of R1 should be $n \times 63$ and R2 contains all integers between 1 and n. Then the JPEG image with the permuted blocks is generated.

3.1.5. Hiding Location Information in the Transformed Image

In this phase, histogram modification is applied to AC coefficients with value 1 or −1 for reversible data embedding (e.g., [15]). The other AC coefficients are modified by histogram shifting. The operation is to embed a bit value i into an original AC coefficient a_j by

$$a_j' = \begin{cases} 0, & a_j = 0 \\ -1, & a_j = -1, i = 0 \\ -2, & a_j = -1, i = 1 \\ 1, & a_j = 1, i = 0 \\ 2, & a_j = 1, i = 1 \\ a_j + 1, & a_j > 1 \\ a_j - 1, & a_j < -1 \end{cases}, \tag{8}$$

where $a_j^{'}$ is the modified AC coefficient.

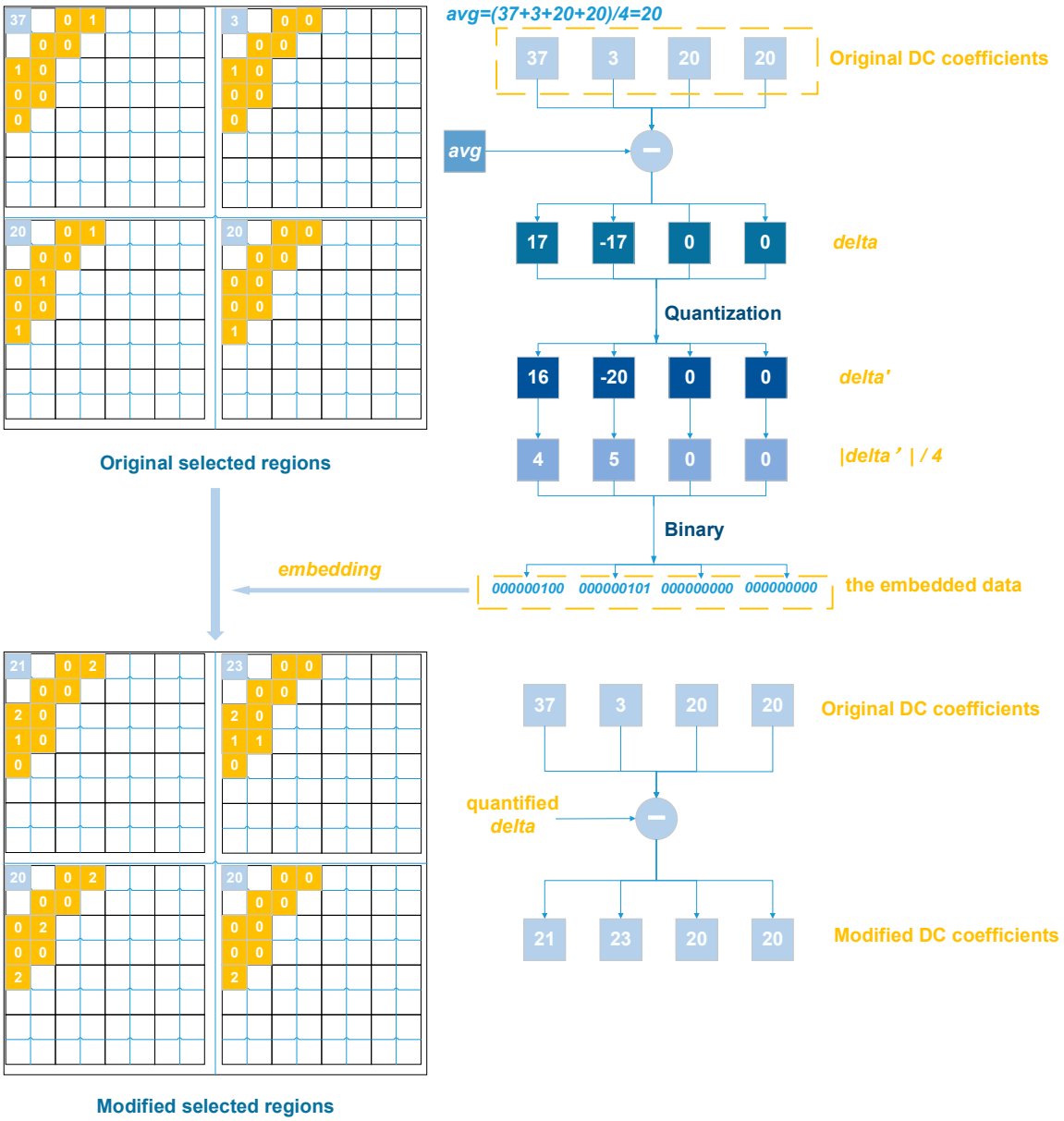

Figure 2. Examples of modifying DC coefficients in selected blocks.

3.2. Recovering the Original JPEG Images

The flowchart of recovering the original JPEG image is illustrated in Figure 3. Note that the same key generation process as in the visual transformation process is conducted to obtain the seed of PRG to generate two random sequences. After extracting location information from the transformed JPEG

image, positions of the selected blocks can be identified. Then the permuted blocks are moved back to their original positions according to the random sequence generated with the secret key. Meanwhile, another random sequence of 1 s and −1 s can be generated as in the visual transformation process so that the flipped AC coefficients can be restored. After that, the original DC coefficient in each block is obtained by extracting the difference value hidden in the AC coefficients. Thus, the original JPEG image is recovered.

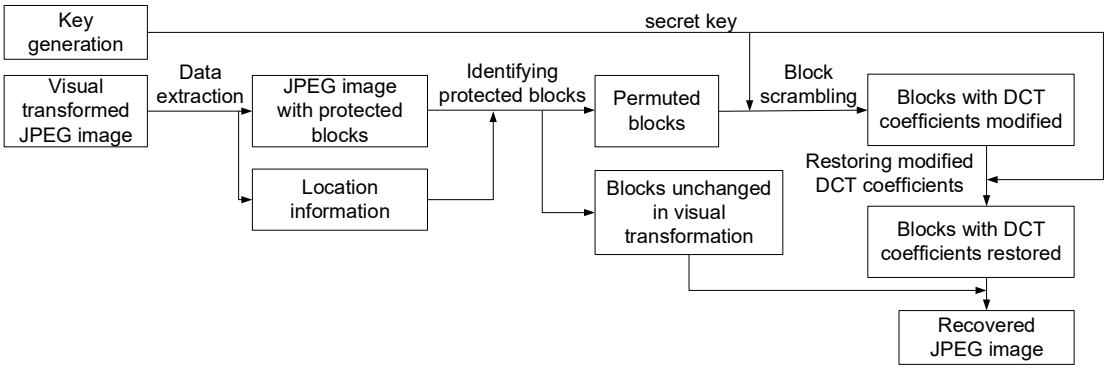

Figure 3. A flowchart of recovering the original JPEG image from the visually transformed one.

3.2.1. Extracting Location Information

According to Equation (8), an AC coefficient contains an embedded bit only when it is ±1 or ±2. So, the coefficients with the embedded data can be distinguished from the others. A hidden bit is extracted from an AC coefficient, while the AC coefficients are restored by

$$i = \begin{cases} 1, & a_j' = -2 \\ 0, & a_j' = -1 \\ 0, & a_j' = 1 \\ 1, & a_j' = 2 \end{cases}, \qquad a_j = \begin{cases} 0, & a_j' = 0 \\ -1, & a_j' = -1, -2 \\ 1, & a_j' = 1, 2 \\ a_j' + 1, & a_j' < -2 \\ a_j' - 1, & a_j' > 2 \end{cases} \tag{9}$$

where i is an extracted bit value, a_j is a restored AC coefficient and $a_j^{'}$ is the corresponding AC coefficient in a visually transformed JPEG image. After data extraction, the values of l, l_1, l_2 in Equations (1)–(3) are calculated by collecting the extracted bits so that the positions of the protected regions can be known.

3.2.2. Recovering the AC Coefficients

According to the secret key, the same seed used in the visual transformation process can be generated and used as the input of PRG to generate the random sequences. Consequently, the location of each permuted block can be known so that each block is moved back to its original position. Note that DC coefficients in the permuted blocks are unmoved in the visual transformation process. Then the signs of all AC coefficients in the restored blocks are flipped with another random sequence of 1 s and −1 s so that the JPEG image with modified DC coefficients is obtained.

3.2.3. Recovering the Modified DC Coefficients

After recovering the AC coefficients, the data hidden in them (i.e., $\frac{|delta'|}{4}$) can be extracted to recover the original DC coefficients. For each block, the embedding data can be extracted by

$$a_j = \frac{a_j'}{2} \tag{10}$$

$$s = a_j' \ mod \ 2, \tag{11}$$

while $a_j^{'}$ ($j = 3, 4, 5, \ldots, 11$) is an AC coefficient carrying a bit value, a_j is the original AC coefficient and s is a bit value in $\frac{|delta'|}{4}$ of the DC coefficient. Then the difference between the modified DC coefficient and the original *delta* can be obtained by

$$delta = \begin{cases} \frac{|delta'|}{4} \times 4 & , \quad if \ \frac{|delta'|}{4} \ mod \ 2 = 0 \\ -\frac{|delta'|}{4} \times 4, & if \ \frac{|delta'|}{4} \ mod \ 2 = 1 \end{cases}. \tag{12}$$

Finally, the DC coefficient in each selected block is restored with the original *delta* by

$$d = d' + delta, \tag{13}$$

where d represents the restored DC coefficient and d' represents the DC coefficient in a visually transformed image. The aforementioned process can be illustrated with the examples in Figure 4.

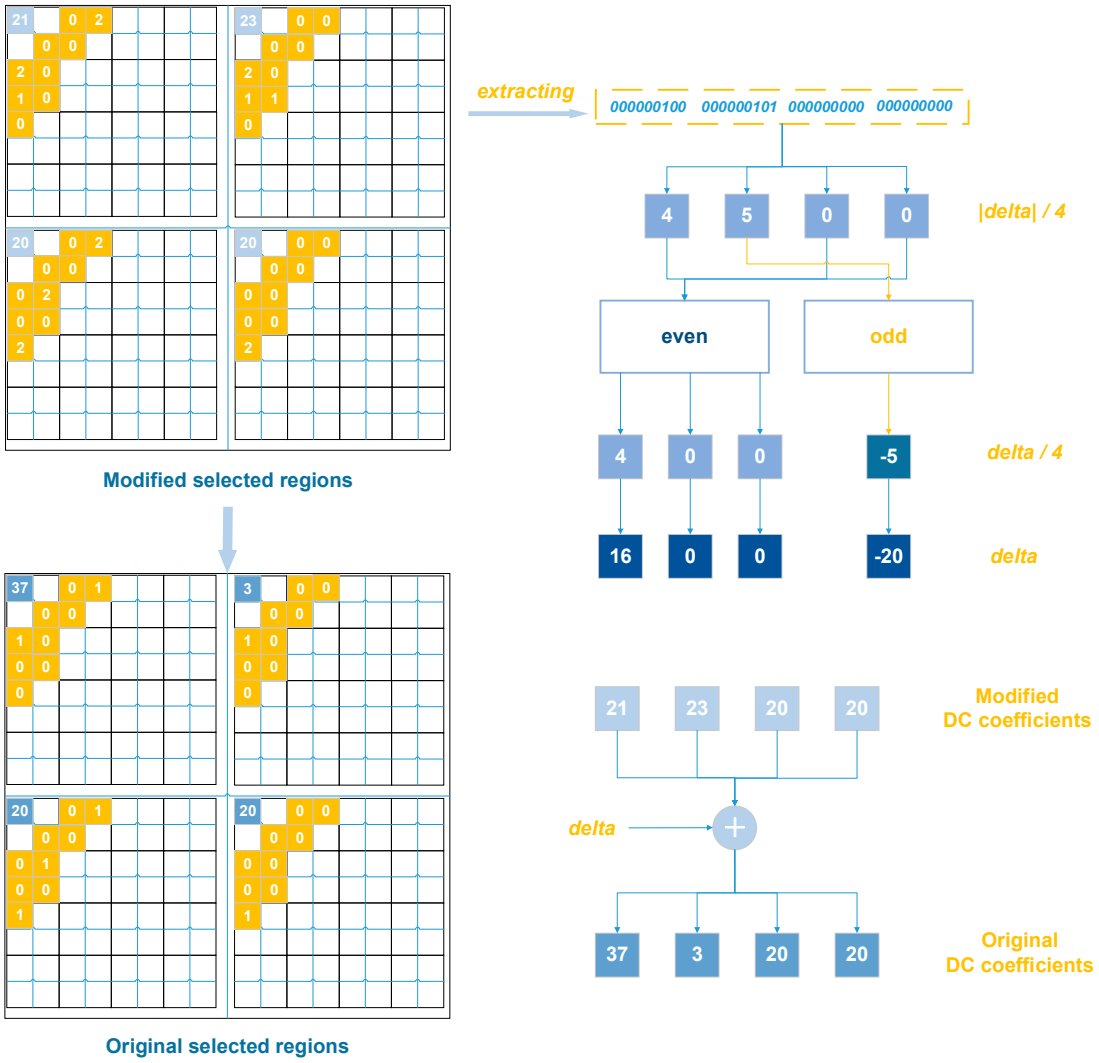

Figure 4. Examples of recovering DC coefficients in selected blocks.

Since one modified DC coefficient is 21 and the extracted $\frac{|delta'|}{4}$ is 4, we know the value of $|delta'|$ is 16. Since $\frac{|delta'|}{4}$ is even, *delta* should be non-negative. So, the value of *delta* is 16 and the original DCT coefficient can be recovered as 37 by adding it to 21. For another DC coefficient 23, the extracted data value is 5, which is an odd number, so *delta* should be negative according to Equation (12) and the

value is −20. The original DCT coefficient is recovered by adding 23 to −20 (i.e., *delta*) to generate 3, which is the original DC coefficient value.

4. Experimental Results

In the experiments, test JPEG images were downloaded from the Images of Groups Dataset [25], which were collected from daily life. All experiments were implemented in the MATLAB environment. Three types of experiment were conducted to evaluate the proposed algorithm, including: (1) visual effects, quality of the transformed images and reversibility, (2) JPEG image file size and (3) security analysis.

4.1. Visual Effects, Image Quality and Reversibility

In this part, the proposed algorithm is evaluated in terms of the invisibility of the protected regions, quality of the unprotected blocks and reversibility. Four testing JPEG images used in the experiments and those with the selected regions protected are shown in Figure 5. In each test image, one or more specific regions were selected and the protected content was made visually invisible while the quality of the unprotected regions remained unchanged. The images recovered with the correct key were identical to the original ones, while the regions obtained with the wrong key were meaningless, as shown in Figure 6.

(**a**) Original image (**b**) Region protected (**c**) Original image (**d**) Region protected

(**e**) Original image (**f**) Region protected (**g**) Original image (**h**) Region protected

Figure 5. Four JPEG images and the ones with user-selected regions protected.

4.2. Change of JPEG Image Size

To evaluate the performance, the proposed algorithm was compared with Yuan's algorithm [6], Li's algorithm [7], P3 [11] and Scrambling JPEG [12] in terms of file size increment after visual transformation. In total one hundred JPEG images with different sizes were employed for testing. Table 1 shows the average JPEG file size increment with the five algorithms, respectively. The file size increment was computed by comparing the file size of a visually transformed image with the original one. The statistical results listed in Table 1 indicate that the proposed algorithm had the smallest increment in file size, which was 0.39% on average for 100 test images. In contrast, Yuan's algorithm and P3 significantly increased the file size. It can be seen that the proposed algorithm was applied with a minimal file size increment of the JPEG images.

(**a**) Recovered with the correct key.

(**b**) Obtained with the wrong key.

Figure 6. JPEG images obtained from the visually transformed ones.

Table 1. Comparisons between five algorithms for content protection in JPEG images.

Algorithm Compared	JPEG File Size Increment (%)
Proposed algorithm	0.39
Yuan's algorithm [6]	10.80
Li's algorithm [7]	0.51
P3 [11]	7.5 ± 2.5
Scrambling JPEG [12]	2.04

4.3. Security Analysis

The security of the proposed algorithm was analyzed by considering a brute force attack [22]. We conducted an experiment on one hundred JPEG images collected from the Images of Groups Dataset [25]. The block number of the protected regions in each JPEG image was calculated and is visualized in Figure 7. It can be seen that the block number ranged from 25 to 1600, while most of them were concentrated between 25 and 400. Besides, the experiments in [13] show that there are 26 nonzero AC coefficients in a block on average. Suppose that the attacker knows the location of the protected regions and the algorithm adopted for visual transformation. An exhaustive search of all the possible combinations of the permuted blocks and the flipped AC coefficients might be conducted. Given that there are at least 25 blocks in a protected region and 26 nonzero AC coefficients in a block, the number of combinations to obtain the permuted blocks is 25!, while there are $2^{26\times25}$ combinations to recover the flipped AC coefficients. Thus, the number of combinations to recover the original image is $25! \times 2^{26\times25}$, which is close to 2^{734}. It can be concluded that the proposed algorithm can resist brute force attack.

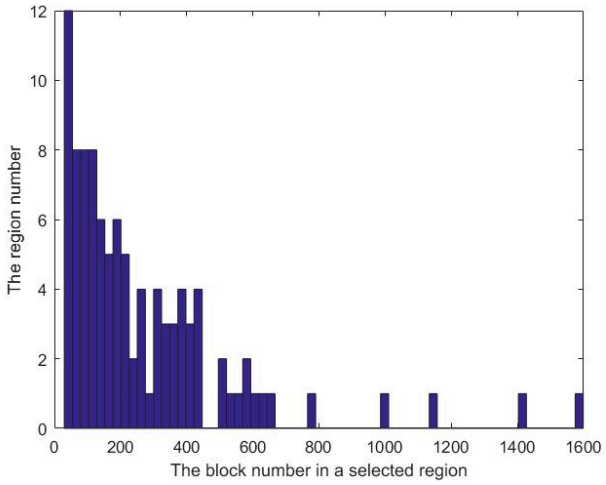

Figure 7. The histogram of the block number in the protected regions.

5. Concluding Remarks

In this paper, a reversible content and privacy protection algorithm has been proposed for JPEG images. Specifically, reversible data hiding and block scrambling have been combined to modify the DCT coefficients in the selected regions. Besides the invisibility of the protected content, reversibility of the transformation process has been achieved. The experimental results have shown that the proposed algorithm has good performance in terms of reversibility, invisibility of the user-selected region, quality of the transformed image and JPEG file size preservation. In addition, the proposed algorithm can resist a brute force attack against content and privacy protection in JPEG images.

Our future work is two-fold: (1) we will study how to improve the visual quality of a visually transformed JPEG image and (2) how to reduce the chance that a protected JPEG image attracts the suspicions of attackers will also be studied.

Author Contributions: Conceptualization, H.-T.W.; Data curation, X.C.; Formal analysis, Y.-m.C.; Funding acquisition, H.-T.W. and Y.-m.C.; Investigation, Y.H. and X.C.; Methodology, Y.H. and X.C.; Project administration, H.-T.W. and Y.-m.C.; Resources, Y.H. and X.C.; Software, Y.H. and X.C.; Writing—original draft, X.C.; Writing—review & editing, H.-T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the Guangdong Province Key Area R&D Program of China with grant number 2019B010137004, in part by the Natural Science Foundation of China with grant numbers 61772208 and 61632013, in part by Hong Kong Baptist University under Grant RC-FNRA-IG/18-19/SCI/03 and RC-IRCMs/18-19/SCI/01, and in part by the Fundamental Research Funds for the Central Universities of China under Grant x2js-D2190700.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Wallace, G.K. The JPEG still picture compression standard. *IEEE Trans. Consum. Electron.* **1992**, *38*, xviii–xxxiv. [CrossRef]
2. Norcen, R.; Uhl, A. Selective encryption of the JPEG2000 Bitstream. In Proceedings of the Communications and Multimedia Security, Advanced Techniques for Network and Data Protection, CMS2003, LNCS, Torino, Italy, 2–3 October 2003; pp. 194–204.
3. Korshunov, P.; Ebrahimi, T. Using face morphing to protect privacy. In Proceedings of the 2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance, Krakow, Poland, 27–30 August 2013; pp. 208–213.
4. Korshunov, P.; Ebrahimi, T. Using warping for privacy protection in video surveillance. In Proceedings of the 2013 18th International Conference on Digital Signal Processing, Fira, Greece, 1–3 July 2013; pp. 1–6.

5. Ong, S.Y.; Wong, K.S.; Tanaka, K. Scrambling–embedding for JPEG compressed image. *Signal Process.* **2015**, *109*, 38–53. [CrossRef]
6. Yuan, L.; Ebrahimi, T. Image transmorphing with JPEG. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 3956–3960.
7. Li, W.; Ni, R.; Zhao, Y. JPEG photo privacy-preserving algorithm based on sparse representation and data hiding. In *International Conference on Image and Graphics*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 575–586.
8. Ma, X.; Yang, L.T.; Xiang, Y.; Zeng, W.; Jin, H. Fully reversible privacy region protection for cloud video surveillance. *IEEE Trans. Cloud Comput.* **2017**, *5*, 510–522. [CrossRef]
9. Joshi, V.B.; Raval, M.S.; Kuribayashi, M. Reversible data hiding based compressible privacy preserving system for color image. *Multimed. Tools Appl.* **2018**, *77*, 16597–16622. [CrossRef]
10. Niimi, M.; Masutani, F.; Noda, H. Protection of privacy in JPEG files using reversible information hiding. In Proceedings of the 2012 International Symposium on Intelligent Signal Processing and Communications Systems, Taipei, Taiwan, 4–7 November 2012; pp. 441–446.
11. Ra, M.-R.; Govindan, R.; Ortega, A. P3: Toward privacy-preserving photo sharing. In Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation, Lombard, IL, USA, 2–5 April 2013; pp. 515–528.
12. Yuan, L.; Korshunov, P.; Ebrahimi, T. Secure JPEG scrambling enabling privacy in photo sharing. In Proceedings of the 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, Ljubljana, Slovenia, 4–8 May 2015; pp. 1–6.
13. Ruchaud, N.; Dugelay, J.L. JPEG-based scalable privacy protection and image data utility preservation. *IET Signal Process.* **2018**, *12*, 881–887. [CrossRef]
14. Coltuc, D. Improved capacity reversible watermarking. In Proceedings of the 2007 IEEE International Conference on Image Processing, San Antonio, TX, USA, 16 September–19 October 2007; pp. 249–252.
15. Huang, F.; Qu, X.; Kim, H.J.; Huang, J. Reversible data hiding in JPEG images. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 1610–1621. [CrossRef]
16. Wu, H.; Tang, H.; Wang, J. A reversible visual transformation algorithm for JPEG images. *J. South China Univ. Technol. (Nat. Sci. Ed.)* **2018**, *46*, 16–21.
17. Daemen, J.; Rijmen, V. *The Design of Rijndael: AES—The Advanced Encryption Standard*; Springer: Berlin/Heidelberg, Germany, 2002.
18. Ciftci, S.; Akyuz, A.O.; Ebrahimi, T. A reliable and reversible image privacy protection based on false colors. *IEEE Trans. Multimed.* **2017**. [CrossRef]
19. Westfeld, A. F5—A steganographic algorithm: High capacity despite better steganalysis. In Proceedings of the International Workshop on Information Hiding, Pittsburgh, PA, USA, 25–27 April 2001. [CrossRef]
20. Wu, H.; Huang, J. Secure JPEG steganography by LSB+ matching and multi-band embedding. In Proceedings of the 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 2737–2740.
21. Rivest, R.L. *The MD5 Message-Digest Algorithm*; RFC Editor; MIT Laboratory for Computer Scienceand RSA Data Security, Inc.: Cambridge, MA, USA, 1992.
22. Brute Force, Wikipedia. Available online: http://en.wikipedia.org/wiki/Brute_force_attack (accessed on 22 September 2020).
23. Hou, D.; Zhang, W.; Yu, N. Image camouflage by reversible image transformation. *J. Vis. Commun. Image Represent.* **2016**, *40*, 225–236. [CrossRef]
24. Wang, K.; Lu, Z.; Hu, Y. A high capacity lossless data hiding scheme for JPEG images. *J. Syst. Softw.* **2013**. [CrossRef]
25. Gallagher, A.; Chen, T. Understanding groups of images of people. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 256–263.