International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161
©2021 INPRESSCO®, All Rights Reserved
Available at http://inpressco.com/category/ijcet

Research Article

Reversible Data Hiding using Color Partitioning Technique for Encrypted Palette Images

Sneha S. Walke and Vandana Navale

Computer Department, Dhole Patil College of Engineering, Wagholi India

Received 10 Nov 2020, Accepted 10 Dec 2020, Available online 01 Feb 2021, **Special Issue-8 (Feb 2021)**

Abstract

Many reversible data hiding schemes have been proposed and successfully applied in medical and military applications. These schemes are developed to ensure digital images authenticity and integrity without any distortion on the original images. The existing RDH techniques are designed for grayscale images and thus, cannot be directly applied to palette images. Since the pixel values in a palette image are not the actual color values, but rather the color indexes. Therefore RDH in the encrypted palette images is more challenging than that designed for the grayscale images. There is no suitable RDH scheme designed for encrypted palette images that has been reported, while palette images have been widely utilized. This has motivated us to design a reliable RDH scheme for encrypted palette images. The proposed method uses a color partitioning method to use the palette colors to construct a certain number of embeddable color-triples, their indexes are self-embedded into the encrypted image so that a data hider can collect the usable color-triples to embed the secret data. For a receiver, the embedded color-triples can be determined by verifying a self-embedded checkcode that enables the receiver to retrieve the embedded data only with the data hiding key. By using the encryption key, the receiver can roughly reconstruct the image content.

Keywords: *Color Partitioning, Image encryption, Information hiding, Image recovery, Palette, Reversible data hiding,*

1. Introduction

Reversible data hiding is a technique to embed additional message into some unacceptable or distorted cover media, such as military or medical images, with a reversible manner so that the original cover content can be perfectly restored after extraction of the hidden message. In recent years the various number of reversible data hiding methods have been proposed. Encryption is an effective and popular means of privacy protection. In order to securely share a secret image with other person, a content owner may encrypt the image before transmission. In some application scenarios, an inferior assistant or a channel administrator hopes to append some additional message, such as the origin information, image notation or authentication data, within the encrypted image though he does not know the original image content. For example, when medical images have been encrypted for protecting the patient privacy, a database administrator may aim to embed the personal information into the corresponding encrypted images. It may be also possible that the original content can be

2. Literature Survey

A. RDH-EI for Uncompressed Images

RDH-EI was used in uncompressed nature images [3]. The owner encrypts the original image using a stream cipher algorithm and then uploads the encrypted image onto the cloud. The cloud server embeds additional bits into cipher text by flipping three least significant bits of half pixels in each block. At the recipient side, the authorized user decrypts the marked(data hidden) encrypted image and generates two candidates for each block by flipping LSBs again. Since the original block of a image is smoother than the interfered block, one hidden bit can be extracted and the original block can be recovered. This method was improved in [5] using a side match algorithm to investigate the spatial correlation between neighboring blocks. The flipping based approaches in [3] and [5] were improved in [6] to reduce errors by comparing more neighbor pixels. However, when the pixels in a block have identical values, data extraction and image recovery in [3-6] may fail. A swapping and shifting based RDH-EI method was proposed to overcome this drawback [7]. Machine learning was also used in RDH-EI to improve the embedding capacity [8]. Although the methods in [2-9] have good embedding and recovery capabilities, data extraction must be done after image decryption. Separable algorithms were proposed to resolve the problem by using High-capacity reversible data hiding in encrypted images by prediction error

[10]. Another type of RDH-EI is realized by preprocessing the original image. This method require the image owner to vacant the space in the plaintext image before image encryption. This additional operation is called as vacating spare rooms in plaintext as preprocessing. Afterwards the image owner encrypts the processed image and uploads it onto the server. Image encryption should not be accounted as preprocessing, since encryption is required in all RDH-EI methods. In [11], image transformation was proposed to encrypt one image to another. Additional bits are embedded by RDH in plaintext images. The preprocessing based methods in [911] can achieve much better embedding rates, but require extra RDH operations before image encryption.

B. RDH-EI for JPEG Bitstreams

The aforementioned RDH-EI is for uncompressed images. However, those methods are not useful in many applications because most images transmitted over Internet are compressed, e.g., the popular JPEG. As a consequence, some RDH-EI works were proposed for JPEG bitstreams [12-15]. The first RDH-EI for JPEG bitstreams was proposed in [12]. This scheme begins with a JPEG encryption algorithm, in which the appended bits of Huffman codes are encrypted by a stream cipher, and all Huffman codes are kept unchanged. After encryption, the

JPEG file size is preserved and the format is compliant to JPEG decoders. On cloud, the server selects the encrypted bitstreams of some blocks as candidates. Additional bits are encoded by LDPC-based error correction codes (ECC) and embedded into the useful candidate bitstream by flipping the LSBs of the encrypted appended bits of the AC coefficients in each candidate block. After the authorized user downloads and decrypts the marked encrypted bitstream, LSBs of the appended bits of useful candidates are flipped again to estimate the additional bits using a predefined blocking artifact function and an ECC decoder. Meanwhile, the original bitstream are losslessly recovered according to the extracted bits. This method is improved in [13], in which the embedding room was reserved before bitstream encryption. Although the embedding capacity is larger, the preprocessing requires the image owner to do an additional computation. Another solution of reversibly hiding data in encrypted JPEG bitstream was proposed in [14], in which image encryption and data embedding are combined together. By scrambling the JPEG structure, the image is encrypted and the additional bits are embedded. The method in [14] cannot be used in cloud storage since the server is unable to embed bits into the encrypted bitstream. To enhance the security of JPEG encryption and improve the embedding capability, another RDH-EI for JPEG bitstream was proposed in [15]. During JPEG bitstream encryption, a new JPEG bitstream is constructed by selecting a portion of blocks from the whole image. Bitstreams of the rest blocks are encrypted and hidden in the JEPG header. With a compression algorithm, some appended bits of the encrypted JPEG bitstream are compressed to accommodate additional bits. On the recipient side, the authorized user uses an iterative algorithm to recovery the original JPEG bitstream according to the variation of blocking artifacts. Compared with [12], this method has larger capacity and better security, and the embedded bits can be directly extracted by the server. Although RDH-EI methods for JPEG in [12-15] have good capabilities in data hiding and image recovery, there exists one problem that the recipient must do a recovery task after decryption. This recovery burden on the recipient side limits the real application of RDH-EI techniques, since users always hope to do nothing except image encryption and decryption. To this end, this paper proposes a new JPEG RDH-EI framework, in which no recovery task is required for the owner or the user. The new RDH-EI[16] framework focuses on labeling the encrypted JPEG images on cloud storage. There are three parties, including the image owner, the cloud server and the authorized user. The owner encrypts a JPEG bitstream and uploads it to the cloud. The cloud server embeds additional messages into the encrypted bitstream to generate a marked encrypted bitstream. The hidden messages can be extracted from the marked encrypted bitstream. When an authorized requires a downloading operation, the server recovers the original encrypted bitstream. After decryption, the user obtains the original JPEG image. The New RDH-EI for JPEG encryption algorithm is also secure against the cipher text-only attack and this works only with the gray scale images

3. Proposed Methodology

We will describe our RDH algorithm with the following three aspects: 1) The encrypted image generation. 2) Data hiding in the encrypted images and 3) Data extraction and image recovery.

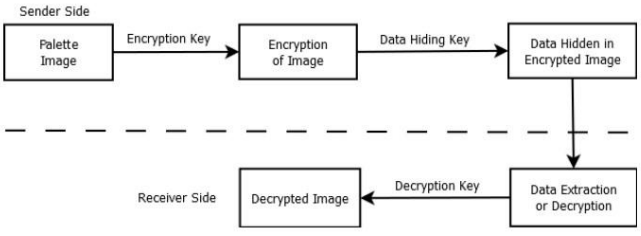

Fig: System Architecture

A. Encrypted Image Generation

To generate an embeddable encrypted image, three phases: 1) color partitioning. 2) Image encryption; and 3) Selfembedding, are involved. Let I{P; C} be the palette image, where P is the index matrix sized N×M and C is the color palette sized n×3. Here, n denotes the number of palette colors. At first, we adopt a color partitioning method to divide colors in C into multiple

color-triples, which are then classified into non-embeddable or embeddable. According to a secret key, we encrypt P and C as P_E and C_E. Then, the indexes of the embeddable color-triples are self-embedded into I{P_E; C_E} together with a flipping check-code and a location map.

1. Color Partitioning: To collect color-triples from C, the occurrence frequency of each palette color is determined. To calculate the difference between any two RGB colors c1(r1, g1, b1) and c2(r2, g2, b2), we use the Euclidean distance between c1 and c2 as the measurement. By repeating the above-mentioned procedure, we can finally construct S. It is found that, during the process to obtain S, the original indexposition of every selected color can be recorded so that the original C can be reconstructed. It indicates that the color partitioning method does not change the set I{P; C}. To enable a data hider to hide secret data in the encrypted domain, we divide the color-triples as embeddable or non-embeddable.

2. Image Encryption: After obtaining S_E, C is encrypted with a cryptographic algorithm, where K_{EC} is the secret key for encrypting C. We compute the bit-stream of each C[k] ($0 \le k \le n$) at first, if C[k] is a 24-bit RGB color (12, 7, 32), the bit-stream will be (00001100, 0000 0111, 00100000)2. Then, we concatenate the bitstreams of all C[k] ($0 \le k < n$) to make a larger bitstream, which is then encrypted with a cryptographic algorithm. We then use the permuted sequence to form P_E with an inverse-operation. As the permutation-based encryption for P has a lower security level.

3. Self-embedding: For data hiding in the encrypted domain, we need to reversibly self-embed some auxiliary data into I{P_E; C_E}. It enables a data hider to use S_E to hide secret data. Though (C[*dhk*], C[*dak*], C[*dbk*]) is encrypted as (C_E[*dhk*],CE[*dak*], C_E [*dbk*]), they both correspond to (*chk*, *cak*, *cbk*) as they have the identical index-positions. The index-positions (*dhk*, *dak*, *dbk*), ($0 \le k < L$), will be self-embedded into P_E. Let *A*IP be the index-information to be embedded. We employ (*ch*0, *ca*0, *cb*0), i.e., (C_E [*dh*0], C_E [*da*0], C_E [*db*0]), to embed *A*IP as it can provide the largest payload. It is noted that, for natural images, it is enough to carry *A*IP by using (*ch*0, *ca*0, *cb*0) according to our experiments. It is straightforward to apply more color-triples if (*ch*0, *ca*0, *cb*0) cannot carry *A*IP. In order to obtain data extraction and image recovery, some additional auxiliary data will be embedded as well. Tthe self-embedded information contains three parts: *A*IP, a flipping check-code *A*FC, and a location map *A*LM. Specifically, $\lceil \log_2 n3 \rceil + 3L \cdot \lceil \log_2 n \rceil$ bits are required to store *A*IP, where $\lceil \log_2 n3 \rceil$ bits are stipulated to store *L*, and $3L \cdot \lceil \log_2 n \rceil$ bits are used to store (*dhk*, *dak*, *dbk*) for $0 \le k < L$. For each
(C_E [*dhk*], C_E [*dak*], C_E [*dbk*]), ($0 \le k < L$).

B. Data Hiding in Encrypted Images

After acquiring I{P_{ES}; C_E}, a data hider should firstly extract *A*IP, *A*FC and *A*LM from PES so as to embed extra information. It is noted that, like the methods in [10-11], it essentially requires a user-server/usermanager interaction [18], i.e., a data hider has to identify the vacated embedding room with the auxiliary data provided by the content owner. All the usable color-triples can be identified with *A*IP. The secret data *M* will be embedded by modifying the pixels of these usable color-triples, which is similar to the selfembedding process. Suppose that, we have extracted *A*IP, *A*FC, and *A*LM from I{P_{ES}; C_E}. It implies that, we can compute (*dhk*, *dak*, *dbk*) for $0 \le k < L$. In the following, we orderly process each color-triple until *M* is fully carried.

Data Extraction and Image Recovery

1. Data Extraction Only with Data Hiding Key:With I(P_{ESM}; C_{EM}) and *k*D, a receiver can extract *M*. It can be performed with an inverse-operation of the data embedding process. At first, we extract (*A*IP, *A*FC, *A*LM) from PESM. Then, according to *A*IP, we reconstruct (*dhk*, *dak*, *dbk*) for all $0 \le k < L$. Since the bit-stream of a color-triple will be flipped if it is used for data hiding, the data extractor can identify the embedded color-triples by comparing *A*FC with the new XOR values of (C_{EM}[*dhk*], C_{EM} [*dak*], C_{EM} [*dbk*]) for all $0 \le k < L$. The reason is that, if a color-triple was flipped, the new XOR value will be different from the original one.

2. Image Recovery Only with Encryption Key*:* In this case, though the receiver cannot recover the original secret message, he can roughly recover the image content. After extracting (*A*IP, *A*FC, *A*LM), *L*E can be computed. Then, by re-flipping (C_{EM} [*dhk*], C_{EM} [*dak*], C_{EM} [*dbk*]), $0 \le k < L$E, C_E. can be reconstructed from C_{EM}. With K_{EC}, C can be further decrypted from C_E.

3. Data Extraction and Image Recovery with Both Keys:In the case that a receiver has both *k*E and *k*D, he can retrieve the original secret message and reconstruct I{P; C} without error. At first, the receiver retrieves the original secret message from I{P_{ESM}; C_{EM} } by applying the above data extraction procedure. Obviously, *Ok* ($1 \le k < L$E) can be reconstructed as well. According to the above image recovery process, C can be recovered without error, and P_E* can be computed as well. With *Ok* for all $1 \le k < L$E, the receiver can reconstruct P_E from P_E*. By decrypting PE as P, I{P; C} can be finally obtained.

4. Result

The methods introduced [11], can provide a relatively higher capacity than the methods introduced in [4,5] and our method. Though the proposed method has a lower embedding capacity than the methods introduced in [11], the proposed method can maintain

high PSNR values. It indicates that, the proposed method can provide a relatively low distortion to the decrypted and marked image. Comparing with the methods in [5,6] since our method does not rely on pixel correlation in both the data extraction and image recovery, our method does not produce an error rate in the data extraction and image recovery.

Parameters	Data Hiding Techniques				
	Propose System	Difference Expansion	Histogram Shifting	Search Order Coding	Block Match Coding for VQ
EC	Low	Low	Moderate	Moderate	High
BR	High	Moderate	Moderate	Low	Low
ER	Low	Moderate	Moderate	Moderate	Moderate
PNSR Value	≈ 32	≈ 36	≈ 42	≈ 45	≈ 46
Time Consuming	Less	Less	Less	More	More
Complexity	Low	Low	Moderate	High	High

Conclusion

RDH in encrypted images is a trending research topic that has drawn attention because of its practical usages. In recent years, some state-of-the-art works have been reported. However, most work of work with grayscale images, and cannot be directly applied to palette images. Palette images are known to have relatively small size, which can reduce storage space and transmission time. This paper introduces a reliable RDH method for encrypted palette images. The data hider can benefit from the data-embedding space reserved by the color partitioning procedure, and apply the color replacement method to embed the additional data.

References

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

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

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

[4]. Z. Qian, H. Zhou, X. Zhang, and W. Zhang, "Separable reversible data hiding in encrypted JPEG bitstreams," IEEE Transactions on Dependable and Secure Computing, doi: 10.1109 /TDSC.2016.2634161.

[5]. W. Hong, T. Chen, and H. Wu, "An improved reversible data hiding in encrypted images using side match," IEEE Signal Processing Letters, vol. 19, no. 4, pp. 199–202, 2012

[6]. X. Liao, and C. Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," Journal of Visual Communication and Image Representation, vol. 28, pp. 21–27, 2015

[7]. Z. Qian, S. Dai, F. Jiang, and X. Zhang, "Improved joint reversible data hiding in encrypted images", Journal of Visual Communication and Image Representation, vol. 40, pp. 732738, 2016.

[8]. J. Zhou, W. Sun, L. Dong, et al. "Secure reversible image data hiding over encrypted domain via key modulation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 3, pp. 441-452, 2016.

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

[10]. X. Wu, and W. Sun, "High-capacity reversible data hiding in encrypted images by prediction error," Signal processing, vol. 104, pp. 387-400, 2014.

[11]. 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

[12]. 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.

[13]. J. C. Chang, Y. Z. Lu, and H. L. Wu, "A separable reversible data hiding scheme for encrypted JPEG bitstreams," Signal Processing, vol. 133, pp. 135-143, 2017.

[14]. S. Y. Ong, K. S. Wong and K. Tanaka, "Scramblingembedding for JPEG compressed image," Signal Processing, vol. 109, pp. 56-68, 2015.

[15]. Z. Qian, H. Zhou, X. Zhang, and W. Zhang, "Separable reversible data hiding in encrypted JPEG bitstreams," IEEE Transactions on Dependable and Secure Computing, doi: 10.1109 /TDSC.2016.2

[16]. Zhenxing Qian, Haisheng Xu, Xiangyang Luo, Xinpeng Zhang "New Framework of Reversible Data Hiding in

[17]. Encrypted JPEG Bitstreams." IEEE Transactions on Circuits and Systems for Video Technology DOI

[18]. 10.1109/TCSVT.2018.27978

[19]. K. Ma, W. Zhang, X. Zhao, N. Yu and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," IEEE Trans. Inf. Forensics Security, vol. 8, no. 3, pp. 553-562, 2013.

[20]. B. Ou, X. Li and W. Zhang, "PVO-based reversible data hiding for encrypted images," IEEE China SIP, pp. 831835,2015.

[21]. X. Li, W. Zhang, X. Gui and B. Yang, "A novel reversible data hiding scheme based on two-dimensional difference-histogram modification," IEEE Trans. Inf. Forensics Security, vol. 8, no. 7, pp. 1091-1100,2013.

[22]. X. Zhang, Z. Qian, G. Feng and Y. Ren, "Efficient reversible data hiding in encrypted images," J. Visual Comm. Image Representation, vol. 25, no. 2, pp. 322-328, 2014.