Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Reversible data hiding scheme for encrypted JPEG bitstreams using adaptive RZL rotation*#

Yongning GUO[1,3], Guodong SU†‡[1,3], Zhiqiang YAO[2], Wang ZHOU[3,4]

[1]*School of Big Data and Artificial Intelligence, Fujian Polytechnic Normal University, Fuzhou 350300, China*

[2]*College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350117, China*

[3]*Engineering Research Center for ICH Digitalization and Multi-Source Information Fusion (Fujian Polytechnic Normal University), Fujian Province University, Fuzhou 350300, China*

[4]*School of Computer and Software Engineering, Xihua University, Chengdu 610039, China*

†E-mail:gdsu@fpnu.edu.cn

Received Nov. 5, 2023; Revision accepted Jan. 22, 2024; Crosschecked Aug. 8, 2024

Abstract: Joint Photographic Experts Group (JPEG) format is extensively used for images in many practical applications due to its excellent compression ratio and satisfactory image quality. Considering compelling concerns about the invasion of privacy, this paper proposes an effective reversible data hiding scheme for encrypted JPEG bitstreams, to provide security and privacy for both secret messages and valuable carriers. First, a format-compatibility and file size preserving encryption algorithm is applied to encipher the plaintext JPEG image into a noise-like version. Then, we present an effective reversible data hiding scheme in encrypted JPEG bitstreams using adaptive RZL rotation, where the secret messages are concealed with the sequence of RZL pairs. When the authorized user receives the marked encrypted JPEG bitstreams, the error-free extraction of secret messages and the lossless recovery of the original plaintext JPEG image can be accomplished separately. Extensive experiments are conducted to show that, compared to some state-of-the-art schemes, the proposed scheme has a superior performance in terms of embedding capacity, while keeping file size preservation and format compatibility.

Key words: Joint Photographic Experts Group (JPEG); Reversible data hiding; Embedding capacity; File size preservation; Format compatibility

https://doi.org/10.1631/FITEE.2300749 **CLC number:** TP39

1 Introduction

Digital image encryption (Pareek et al., 2006; Xie et al., 2017; Zhang LY et al., 2018; Chuman et al., 2019; Hua et al., 2019) is a technique that has been designed to encrypt a plaintext image into an encrypted version, from which no meaningful information is revealed. In this way, the security and privacy of the plaintext image are guaranteed so that the encrypted images can be uploaded onto a cloud storage server without worry. Different from image encryption, reversible data hiding (RDH) is a technique to imperceptibly embed secret messages into a plaintext image. Correspondingly, the plaintext image can be losslessly restored after removing the hidden secret messages from the stego-image. Existing RDH schemes can be classified roughly into five categories: lossless compression (Ni et al., 2006; Zhang WM et al., 2013; Qin

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 62272103), the Natural Science Foundation of Fujian Province, China (Nos. 2021J011237, 2022J01971, 2022J01972, 2022J01974, and 2022J01975), and the Open Fund of Engineering Research Center for ICH Digitalization and Multi-Source Information Fusion (Fujian Polytechnic Normal University), China (Nos. G3-KF2202 and G3-KF2205)

Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2300749) contains supplementary materials, which are available to authorized users

ORCID: Guodong SU, https://orcid.org/0000-0002-3050-7166

and Hu, 2016), difference expansion (Tian, 2003; Wang et al., 2017; Su et al., 2019), prediction error expansion (Ou et al., 2013; He WG et al., 2018; Bai et al., 2021), histogram shifting (HS) (Li et al., 2013; Ying et al., 2019; Peng et al., 2020), and information encoding (Chang CC et al., 2018, 2019; Chang CC and Li, 2019). Due to the reversible property, the RDH scheme is used extensively in practical applications, e.g., military communication and medical diagnoses.

RDH schemes are able to ensure the confidentiality of secret messages and have the property of reversibility. However, they are poor in terms of ensuring the security of plaintext images. Therefore, many researchers are interested in developing RDH technique for encrypted images (RDHEI). Generally speaking, an RDHEI scheme consists of three parties, the image owner, data hider, and authorized user. The image owner first enciphers a plaintext image into an encrypted version and transmits it to the data hider, and then the data hider hides messages within it. From the authorized user side, the hidden messages can be extracted error-free, the plaintext image can be recovered completely, or both. To the best of our knowledge, most RDHEI schemes are designed to work for uncompressed images and can be separated into the following categories: vacating room before encryption (Ma et al., 2013; Puteaux and Puech, 2018; Qiu et al., 2020), vacating room after encryption (Zhang XP, 2012; Wu et al., 2019), and vacating room by specific encryption (Xiao et al., 2017; Qin et al., 2018; Yi et al., 2018; Huang DL and Wang, 2020; Mohammadi et al., 2020; Qi et al., 2023; Su and Chang, 2023; Zhou et al., 2023). Such schemes provide a considerable embedding rate, with a value of an approximately equal to or higher than 1.0 bpp.

Meanwhile, images in compressed format are desired in many applications to save transmission time and reduce bandwidth and storage costs. Among those, Joint Photographic Experts Group (JPEG) (Rabbani and Joshi, 2002; Huang FJ et al., 2015; He JH et al., 2018, 2020) is one of the most commonly used formats of lossy compression for digital images since it provides a considerable compression ratio and a satisfactory image quality. More recently, research has been undertaken on how to enhance the embedding capacity in encrypted JPEG images (Qian et al., 2014, 2018, 2019; Chang JC et al., 2017; He JH et al., 2019; Puteaux et al., 2021; Hua et al., 2023; Yuan et al., 2023). Further details of these schemes will be introduced and analyzed in Section 2.2.

In this paper, we present a high capacity RDH scheme in encrypted JPEG bitstreams to address the concerns about privacy information protection. The contributions and novelties of this paper are summarized as follows:

1. We propose an effective RDH scheme in encrypted JPEG bitstreams based on RZL (run size of zeros/level of a non-zero alternating current (AC) coefficient) rotation. First, an effective encryption algorithm is exploited to encipher JPEG bitstreams while keeping format compatibility and file size preservation well. Then, RZL pairs from a discrete cosine transform (DCT) block are parsed from received JPEG bitstreams and form an ordered sequence. Next, we construct a one-to-one mapping relationship between the secret messages and the ordered sequence, so that the secret messages can be simply embedded into the sequence of RZL pairs, rather than into the RZL pairs themselves. Experimental results show that the proposed scheme has a good embedding capacity with the features of file size preservation and format compatibility.

2. The proposed RDH scheme in the encrypted JPEG domain offers a satisfactory embedding capacity with file size preservation and format compatibility compared with other related works. Additionally, data extraction and JPEG image recovery are separable and reversible.

3. We conduct a more comprehensive evaluation, which shows that our proposed scheme provides a considerable security level and a lower computational complexity.

2 Related works

In this section, the simplified syntax of the baseline JPEG is briefly stated (International Telecommunication Union, 1992). Its framework is shown in Fig. 1, and the main terms used in this paper are listed in Table 1.

2.1 JPEG bitstream parsing

In the JPEG encoder, an image with the dimensions of $H\times W$ is first divided into L non-overlapping 8×8 blocks, where $L=hw$, $h=H/8$, and $w=W/8$. Then, each

Table 1 Main terms used in this paper

Acronym	Term	Acronym	Term
JS	JPEG bitstream	DCC	Code of a DC differential coefficient
SOI	Start-of-image marker	ACC	Code of an RZL pair
EOI	End-of-image marker	DCH	Huffman code in a DCC
EOB	End-of-block marker	DCA	Appended bits in a DCC
JH	JPEG header	ACH	Huffman code in an ACC
ECS	Entropy encoded segment	ACA	Appended bits in an ACC

image block is converted into a set of DCT coefficients which are further quantized as quantized DCT coefficients. Next, all quantized DCT coefficients in a block are encoded as an ECS, and ECSs of those L blocks are sequentially cascaded as the entropy encoded data. Finally, the SOI, JH, entropy encoded data, and EOI are combined to form a JPEG bitstream JS, as shown in Fig. 1. Therefore, JS can be represented as

$$\mathrm{JS}=\left\{\mathrm{SOI},\mathrm{JH},\mathrm{ECS}^1,\mathrm{ECS}^2,\cdots,\mathrm{ECS}^L,\mathrm{EOI}\right\}. \quad (1)$$

Moreover, as can be seen from Fig. 1, each ECS contains a DCC, multiple ACCs, and an EOB. Each DCC consists of a DCH and a DCA, and each ACC consists of an ACH and an ACA. To summarize, the ECS in the l^{th} block can be represented by

$$\begin{aligned}\mathrm{ECS}^l&=\left\{\mathrm{DCC}^l,\mathrm{ACC}_1^l,\mathrm{ACC}_2^l,\cdots,\mathrm{ACC}_{K_l}^l,\mathrm{EOB}\right\}\\&=\left\{[\,\mathrm{DCH}^l,\mathrm{DCA}^l\,],[\,\mathrm{ACH}_1^l,\mathrm{ACA}_1^l\,],\right.\\&\quad\left.[\,\mathrm{ACH}_2^l,\mathrm{ACA}_2^l\,],\cdots,[\,\mathrm{ACH}_{K_l}^l,\mathrm{ACA}_{K_l}^l\,],\mathrm{EOB}\right\},\end{aligned} \quad (2)$$

where K_l is the number of ACCs in the l^{th} block (l= 1, 2, ⋯, L).

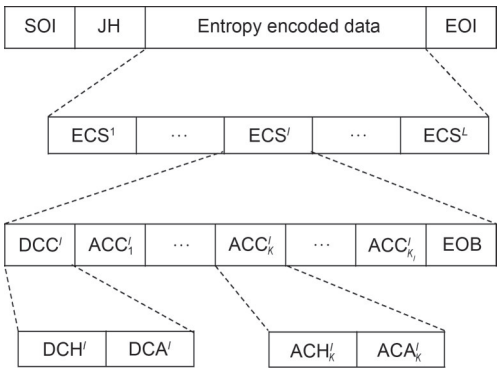

Fig. 1 Simplified syntax of baseline JPEG

2.2 Scheme review and analysis

In the last decade, many schemes have been developed to achieve a good embedding capacity and secure privacy protection. It is also desired that these schemes (Qian et al., 2014, 2018, 2019; Chang JC et al., 2017; He JH et al., 2019; Puteaux et al., 2021; Hua et al., 2023; Yuan et al., 2023) can always ensure both format compatibility and file size preservation even when a legal operation is conducted, such as encryption or steganography.

As far as we known, Qian et al. (2014) are the first to investigate the RDH scheme in encrypted JPEG bitstreams. In their scheme, all the appended bits and the quantization table were encrypted with a standard stream cipher function to distort the JPEG image. Secret messages were encoded with error correction codes and concealed by slightly modifying the least significant bit (LSB) of the appended bits of ACCs within the usable segments. A self-defined block artifact function was used to serve for accurate data extraction and lossless image recovery. To achieve the property of separability, Qian et al. (2018) presented a novel RDH scheme for encrypted JPEG bitstreams. On one hand, the ECSs corresponding to several DCT blocks were randomly selected to generate a new JPEG bitstream with a smaller size, where all appended bits of those ECSs were encrypted using a stream cipher algorithm. On the other hand, all bits of ECSs corresponding to the remaining DCT blocks were conducted in the same way. Using matrix encoding, secret messages were embedded into all encrypted appended bits of ACCs captured from the remaining DCT blocks. Finally, all bits of the remaining DCT blocks were put inside the reserved application segments, e.g., APPn in JH. At the receiver side, both secret messages and plaintext JPEG image can be recovered losslessly.

Chang JC et al. (2017) observed that there is redundancy room for exploitation in compressing the LSBs of the two-bit appended values in the JPEG bitstreams. Therefore, an effective lossless data compression algorithm was designed to vacate space for future secret message embedding. Afterwards, the processed JPEG bitstream was encrypted through block-wise permutation and the secret messages were directly replaced into pre-reserved space using bit replacement. It is worth noting that for their application, a renewed encoding of two-bit appended values is required, which is a little different from other schemes. Qian et al. (2019) developed a novel framework for RDH in encrypted JPEG bitstreams, where the data embedding and extraction were accomplished on the server side. The JPEG encryption scheme they employed is similar to the scheme used by Qian et al. (2014), and the main difference is that, instead of directly using XOR direct current (DC) differential values of selected blocks, their new differential values were re-calculated and encoded by Huffman codes. This also results in a smaller size of the encrypted JPEG image. In the data hiding phase, both Huffman code mapping and the ordered histogram shifting strategies were employed to greatly improve the embedding payload, which is better than that of recent works (Qian et al., 2014, 2018; Chang JC et al., 2017; He JH et al., 2019).

Based on an encryption of He JH et al. (2018)'s scheme, a novel RDH scheme in encrypted JPEG bitstreams with high capacity was proposed by He JH et al. (2019), where secret messages were embedded into encrypted JPEG bitstreams based on invariant zero-run length in the RZL pairs. In this scheme, the secret message extraction and lossless image recovery can be implemented separately. Puteaux et al. (2021) proposed a hierarchical high capacity data hiding approach for JPEG crypto-compressed images using sign bit substitution. The stronger the embedding capacity is, the more the diagonals are used, and the better quality the recovered JPEG image is preserved. Combining the encryption algorithm (He JH et al., 2018) and histogram shifting based RDH, Yuan et al. (2023) presented a JPEG-RDHEI scheme to embed the secret messages into some zero coefficients with a certain file growth. Through changing the ACCs, the chosen plaintext attack was ensured at the expense of abandoning image content based adaptive encryption key. Inspired by He JH et al. (2019), Hua et al. (2023) boosted the number of embeddable blocks using a grouping mechanism and designed a permutation based embedding technique that allows more secret messages. The embedding capacity has gradually been improved for those schemes (He JH et al., 2019; Hua et al., 2023; Yuan et al., 2023), but more iterations are required during the encryption of DCCs to obtain an acceptable visual security. Detailed comparisons of different schemes will be presented in Section 4.3.

3 Proposed RDH scheme in encrypted JPEG bitstreams

In this paper, we present a novel RDH scheme in encrypted JPEG images to further enhance embedding capacity, while always keeping format compatibility and a friendly file size. Fig. 2 shows the basic framework of the proposed scheme. To begin with, the image owner ciphers the plaintext JPEG bitstreams O_{JS} through the joint use of the encryption of DC coefficients and the encryption of ACCs, which are considered to perform well in terms of format compatibility and file size preservation. Then, the image owner uploads the encrypted JPEG bitstreams E_{JS} to the data hider's side, and the data hider conceals the secret messages in E_{JS} to generate the marked encrypted JPEG bitstreams M_{JS}. On the authorized user's side, the secret messages can be extracted error-free and the original plaintext JPEG image can be recovered losslessly when he/she holds the corresponding encryption keys.

3.1 Encryption on JPEG bitstreams

Considering the fact that the encryption algorithms applied in schemes (Qian et al., 2014, 2018, 2019; Chang JC et al., 2017; He JH et al., 2019; Hua et al., 2023; Yuan et al., 2023) lead to overflow of decoded DC coefficients because of their XOR-ing of DCAs directly, an effective encryption method (Su et al., 2023) for plaintext JPEG bitstreams is employed in this paper to blur the plaintext JPEG image. This encryption algorithm is considered to strictly follow the JPEG standard, and the detailed procedures are described as follows:

1. Row-wise shuffling and reversing of DC coefficients. The $h\times w$ DC coefficients collected from

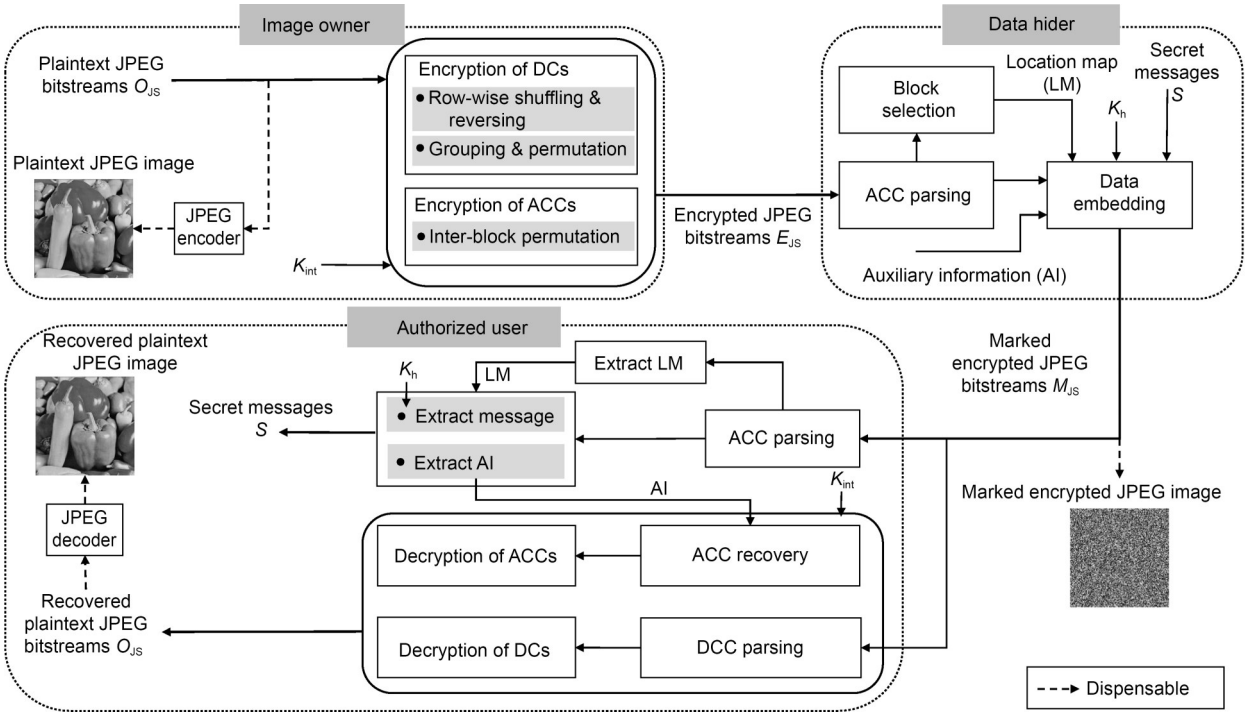

Fig. 2 Basic framework of the proposed scheme

each DCT block are divided into non-overlapping blocks, i.e., B_i ($1 \le i \le h$), with a size of $1 \times w$. $\{B_i\}_{1 \le i \le h}$ is globally shuffled and all coefficients within B_i are alternatively reversed. The processes of shuffling and reversing are decided by the joint use of image features and an initial key K_{int}.

2. Grouping and permutation of DC differential values. First, the above modified DC coefficients are coded using the differential pulse code modulation (DPCM), resulting in a sequence of DC differential values. Then, all DC differential values are adaptively separated into T groups under the circumstance of not causing the overflow of decoded ciphered DC coefficients. Among them, every two adjacent groups can be automatically differentiated by a differential mark. Finally, each group is internally permuted.

3. Encryption of ACCs. To overcome the revelation of the outline of an image, the ECSs excluding DCCs in inter-blocks are shuffled.

3.2 ACC parsing

As we know, the JPEG encoder scans a DCT block's AC coefficients in the zig-zag scanning order and transforms them into a one-dimensional sequence. Next, those AC coefficients are encoded as several pairs, and each of them is described as a tuple, i.e., RZL. For simplicity, we denote the k^{th} RZL in the l^{th} DCT block as

$$\mathbb{Z}_k^l = r_k^l / e_k^l, \tag{3}$$

where r_k^l is the number of consecutive zeros before the k^{th} non-zero AC coefficient e_k^l. Afterwards, the RZLs are coded into bitstreams by employing entropy encoding. This, in turn, implies that it is easy to parse RZLs from the received JPEG bitstreams.

Fig. 3 presents a specific example to illustrate the construction of RZLs. Fig. 3a shows the quantized DCT coefficient matrix, where the first coefficient is the DC coefficient and the others are AC coefficients. The process of RZL pairing is demonstrated in Fig. 3b.

3.3 Adaptive state mapping definition

As we can see from Fig. 3, such a property can be observed in AC coefficients; that is, the low frequency coefficients are larger in terms of absolute magnitude when compared to the high frequency coefficients. Following this fact, an adaptive state mapping is defined and used to conceal secret messages.

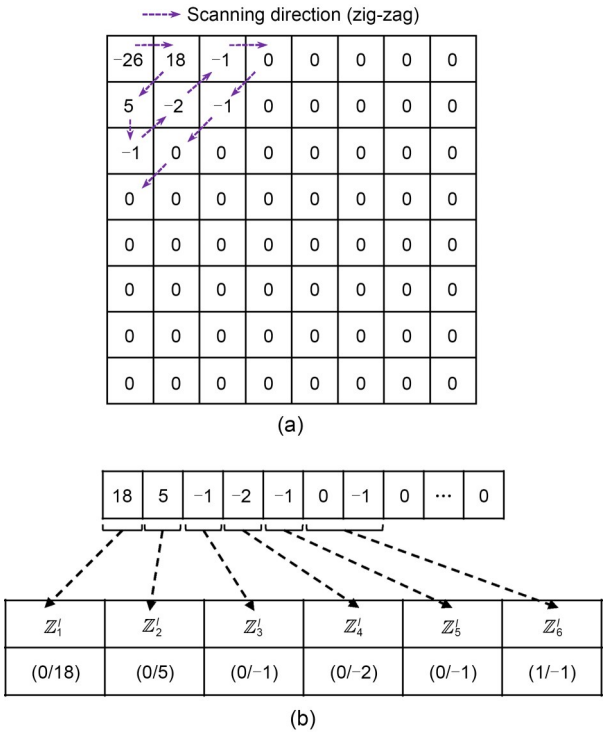

Fig. 3 An example of RZL pairing for the l^{th} block: (a) quantized DCT coefficient matrix; (b) RZL pairing

For the l^{th} block, N RZL pairs, including the first RZL pair ($\mathbb{Z}_1^l$) and the $(K_l-N+1)^{\text{th}}$ to $(K_l-1)^{\text{th}}$ RZL pairs ($\mathbb{Z}_{K_l-N+1}^l$, $\mathbb{Z}_{K_l-N+2}^l$, $\cdots$, $\mathbb{Z}_{K_l-1}^l$), are chosen and combined as a new sequence. For ease of illustration, the selected sequence $\boldsymbol{\theta}^l$ is represented as

$$\boldsymbol{\theta}^l=\left[\mathbb{Z}_1^l,\mathbb{Z}_{K_l-N+1}^l,\mathbb{Z}_{K_l-N+2}^l,\cdots,\mathbb{Z}_{K_l-1}^l\right], \tag{4}$$

where N is ranged in $\left\{2^1,2^2,\cdots,N_{\max}=2^{\lfloor\log_2(K_l-1)\rfloor}\right\}$, and "$\lfloor\ \rfloor$" represents the floor operation. The last pair $\mathbb{Z}_{K_l}^l$ is not used as the selected RZL pair, because it may be employed to conceal the auxiliary information (AI) later, which will be discussed in Section 3.5.

Using $\boldsymbol{\theta}^l$, N distinct ordered sequences $\{\boldsymbol{\theta}_t^l|0\le t\le N-1\}$ can be derived by using the right-oriented rotation, as shown in Fig. 4a. Each ordered sequence $\boldsymbol{\theta}_t^l$ can be used to carry a $(\log_2 N)$-bit secret message s_t^l. Moreover, for an ordered sequence $\boldsymbol{\theta}_t^l$, its feature λ_t^l is defined as

$$\lambda_t^l=\left|e_{\text{first}}^l\right|, \tag{5}$$

where e_{first}^l is the value of the non-zero AC coefficient in the first RZL pair in the current $\boldsymbol{\theta}_t^l$.

Generally, $\lambda_0^l=|e_1^l|$ is expected to be the unique largest one in $\{\lambda_0^l,\lambda_1^l,\cdots,\lambda_{N-1}^l\}$ in most DCT blocks. This is mainly to benefit from the design of the quantizer in the JPEG encoder, where the low frequency coefficients are quantized using smaller divisors and high frequency coefficients are quantized using larger divisors. If a DCT block meets this expectation, it may be used to carry secret messages. To better demonstrate the definition of adaptive state mapping, Fig. 4b provides an example of the construction of a state mapping table for the l^{th} block shown in Fig. 3, where N is set to $2^{\lfloor\log_2(6-1)\rfloor}=4$. As can be seen from Fig. 4b, $\lambda_0^l=|e_1^l|=18$ is much larger than the others; thus, this DCT block is on standby.

3.4 Block selection and location map

For the aim of recovering the original JPEG image losslessly, the DCT blocks are separated into three types, including embeddable blocks, un-embeddable blocks, and invalid blocks, according to the value of K_l and the sequence $\{\lambda_0^l,\lambda_1^l,\cdots,\lambda_{N-1}^l\}$. The definitions of those three types with regard to DCT blocks are described as follows and some examples corresponding to the three types are listed in Fig. 5:

1. Embeddable block. $K_l>2$, $\lambda_0^l>1$, and λ_0^l is the unique maximum value in $\{\lambda_0^l,\lambda_1^l,\cdots,\lambda_{N-1}^l\}$ when $N\in\left\{2^1,2^2,\cdots,N_{\max}=2^{\lfloor\log_2(K_l-1)\rfloor}\right\}$.

2. Un-embeddable block. $K_l>2$, $\lambda_0^l>1$, but the unique maximum value in $\{\lambda_0^l,\lambda_1^l,\cdots,\lambda_{N-1}^l\}$ is not λ_0^l when $N\in\{2^1,2^2,\cdots,N_{\max}\}$.

3. Invalid block. $K_l\le2$, $\lambda_0^l\le1$, or not fewer than two values of $\{\lambda_0^l,\lambda_1^l,\cdots,\lambda_{N-1}^l\}$ are at a maximum when $N\in\{2^1,2^2,\cdots,N_{\max}\}$.

In our proposed scheme, the payload, including AI and secret messages, will be only concealed in the embeddable blocks, so that we can extract secret messages error-free and recover plaintext JPEG images losslessly on the authorized user's side. Nevertheless, it is obvious that an embeddable block may be altered to an un-embeddable block during data embedding. For this, a location map, namely LM, is defined and embedded into JPEG bitstreams along with secret messages so that the authorized user can differentiate the embeddable block and un-embeddable block. Concretely, if a DCT block is defined as the embeddable type, it is marked with bit "1" in LM; if a DCT block is defined as the un-embeddable type, it is marked

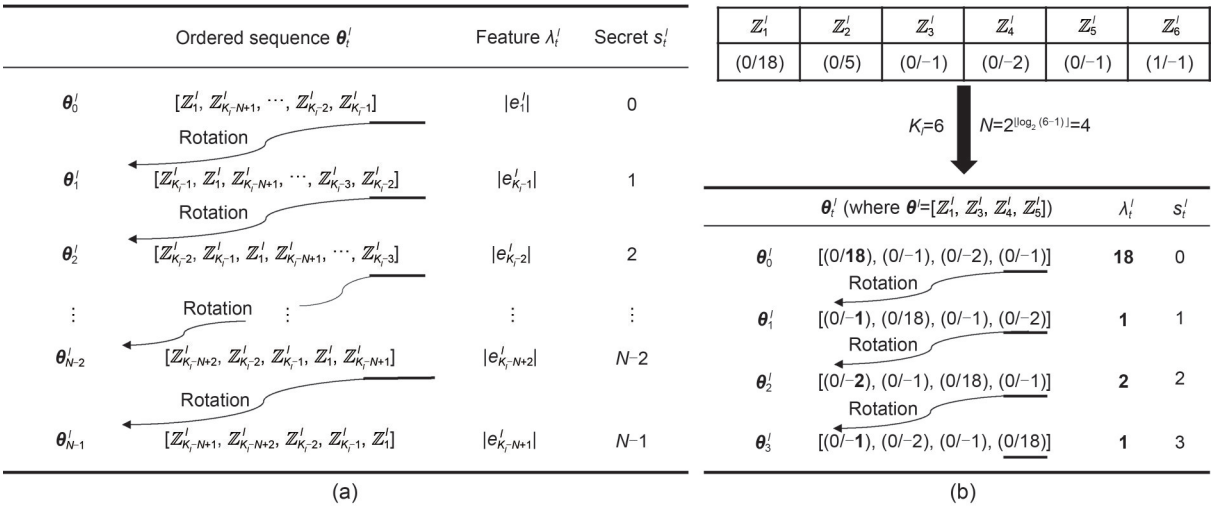

Fig. 4 Adaptive state mapping definition (a) and an example of state mapping table for case in Fig. 3 (b)

RZL pairs in a DCT block		*N* RZL pairs selected	Feature $[\lambda_0^l, \lambda_1^l, \cdots, \lambda_{N-1}^l]$		Valid *N*	Block type
[(0/18), (0/5), (0/−1), (0/−2), (0/−1), (1/−1)]	$N=2^2$	[(0/18), (0/−1), (0/−2), (0/−1)]	[18, 1, 2, 1]	☑	2^2	Embeddable
	$N=2^1$	Ignored	Ignored			
[(0/18), (0/5), (0/−18), (0/−2), (0/−1), (1/2), (2/−2), (0/1), (0/−1)]	$N=2^3$	[(0/18), (0/5), (0/−18), (0/−2), (0/−1), (1/2), (2/−2), (0/1)]	[18, 5, 18, 2, 1, 2, 2, 1]	☒	2^2	Embeddable
	$N=2^2$	[(0/18), (1/2), (2/−2), (0/1)]	[18, 2, 2, 1]	☑		
	$N=2^1$	Ignored	Ignored			
[(0/−2), (1/−1), (2/2), (1/−3), (0/1)]	$N=2^2$	[(0/−2), (1/−1), (2/2), (1/−3)]	[2, 1, 2, 3]	☒	None	Un-embeddable
	$N=2^1$	[(0/−2), (1/−3)]	[2, 3]	☒		
[(0/−2), (2/−1), (1/2), (2/−2), (0/1)]	$N=2^2$	[(0/−2), (2/−1), (1/2), (2/−2)]	[2, 1, 2, 2]	☒	None	Invalid
	$N=2^1$	[(0/−2), (2/−2)]	[2, 2]	☒		
[(0/−1), (3/−1), (0/1)]		$\lambda_0^l \le 1$	Ignored	☒	None	Invalid
[(0/−2), (0/−1)]		$K_l \le 2$	Ignored	☒	None	Invalid

Fig. 5 Examples for identifying three block types

with bit "0" in LM. Obviously, the length of LM, represented by |LM|, is equal to the sum of the number of embeddable blocks and the number of un-embeddable blocks. It is also worth noting that it is not necessary to record the location information of invalid blocks in LM because the types of invalid blocks remain unchanged either before or after data embedding.

3.5 Data embedding

Our data embedding scheme consists of embedding LM and embedding payload. With respect to LM, we insert it into the LSB of $\mathbb{Z}_{K_l}^l$'s of the first |LM| DCT blocks by using bit replacement. To be reversible, the original LSBs of above $\mathbb{Z}_{K_l}^l$'s are collected beforehand as AI. Then, the payload, including AI and encrypted secret (ES), is further inserted into E_{JS}, where ES is the result of XOR-ing the secret messages S using an encryption key K_{h}. The detailed process of data embedding can be described by Algorithm 1.

To better explain the process of data embedding, two examples are in Fig. 6 to illustrate the selected RZL pair rotation and payload embedding.

1. Example in Fig. 6a

Let us assume that the to-be-embedded payload bits are "10" and the original RZL pairs in an embeddable block are [(0/18), (0/5), (0/−1), (0/−2), (0/−1), (1/−1)].

Since its valid N is 4 (Fig. 5), thus the 1st and the 3rd to 5th RZL pairs are selected, i.e., [(0/18), (0/−1), (0/−2), (0/−1)]. Simultaneously, we convert the payload bits "10" into decimal value t=2. Afterwards, the selected RZL sequence [(0/18), (0/−1), (0/−2), (0/−1)] is rotated right t=2 times, resulting in a new temporary RZL sequence [(0/−2), (0/−1), (0/18), (0/−1)]. Finally, the RZL pairs in the corresponding marked embeddable block are replaced as [(0/−2), (0/5), (0/−1), (0/18), (0/−1), (1/−1)]. At the end, the payload bits "10" are carried.

2. Example in Fig. 6b

Take another specific case for example. Suppose that the payload bits are "11" and the corresponding decimal value is t=3. It will be concealed into an embeddable block whose original RZL pairs are [(0/18), (0/5), (0/−18), (0/−2), (0/−1), (1/2), (2/−2), (0/1), (0/−1)], where K_l=9. It can be seen from Fig. 5 that its valid N is 4 not 8 because there exist two maximum feature values in the selected sequence when N=2^3=8. Therefore, the 1st and 6th–8th RZL pairs are determined to form the RZL sequence, i.e., [(0/18), (1/2), (2/−2), (0/1)]. Next, the selected RZL sequence is altered to the state $\boldsymbol{\theta}_3^l$=[(1/2), (2/−2), (0/1), (0/18)] by rotating $\boldsymbol{\theta}_0^l$ right t=3 times, to carry the payload bits "11." In the following, pair replacement is performed to generate the new RZL pairs in a marked embeddable block, i.e., [(1/2), (0/5), (0/−18), (0/−2), (0/−1), (2/−2), (0/1), (0/18), (0/−1)].

Algorithm 1 Data embedding

Input: ES, encrypted JPEG bitstreams E_{JS}

Output: marked encrypted JPEG bitstreams M_{JS}

1 Derive RZL pair sequence $\left[\mathbb{Z}_1^l, \mathbb{Z}_2^l, \cdots, \mathbb{Z}_{K_l}^l\right]$ from E_{JS}

2 Determine block types, valid N, and a location map LM

3 Collect the auxiliary information AI

4 Derive payload P=AI||ES=$(p_1 p_2 \cdots p_{|P|})_2$, where $|P|$ represents the length of P

5 Embed the LM into the LSB of $\mathbb{Z}_{K_l}^l$'s in the first |LM| DCT blocks

6 Embed P into embeddable blocks. For the l^{th} block:

7 **if** block is embeddable

8 Select N pairs $\boldsymbol{\theta}^l = \left[\mathbb{Z}_1^l, \mathbb{Z}_{K_l-N+1}^l, \mathbb{Z}_{K_l-N+2}^l, \cdots, \mathbb{Z}_{K_l-1}^l\right]$ and initialize $\boldsymbol{\theta}_0^l$=$\boldsymbol{\theta}^l$

9 Take $\log_2 N$ bits from the head of P and convert them into the decimal value t

10 Rotate $\boldsymbol{\theta}_0^l$ right t times, resulting in $\boldsymbol{\theta}_t^l$

11 Replace $\boldsymbol{\theta}_0^l$ by $\boldsymbol{\theta}_t^l$ at the same positions in $\left[\mathbb{Z}_1^l, \mathbb{Z}_2^l, \cdots, \mathbb{Z}_{K_l}^l\right]$, resulting in a new $\left[\bar{\mathbb{Z}}_1^l, \bar{\mathbb{Z}}_2^l, \cdots, \bar{\mathbb{Z}}_{K_l}^l\right]$

12 Update P by discarding its first $\log_2 N$ bits

13 **end if**

14 Repeat step 6 until all blocks have been processed

15 Encode $\left[\bar{\mathbb{Z}}_1^l, \bar{\mathbb{Z}}_2^l, \cdots, \bar{\mathbb{Z}}_{K_l}^l\right]$ into the corresponding entropy code for all DCT blocks

16 Output marked encrypted JPEG bitstreams M_{JS}

3.6 Data extraction and image recovery

When the authorized user receives the marked encrypted JPEG bitstreams M_{JS}, he/she can extract the secret messages S successfully and recover the original JPEG bitstreams $\dot{O}_{\mathrm{JS}}$ losslessly.

3.6.1 Extraction of secret messages

If the authorized user owns the hiding key K_{h}, the secret messages can be extracted without error. First, he/she performs ACC parsing for each block to derive RZL pair sequence $\left[\bar{\mathbb{Z}}_1^l, \bar{\mathbb{Z}}_2^l, \cdots, \bar{\mathbb{Z}}_{K_l}^l\right]$ from M_{JS}. Considering that the invalid DCT blocks are always unchanged, they can be identified according to the definition of "invalid block" mentioned in Section 3.4, so that the length of the location map |LM| is determined, which equals the total number of all blocks minus the number of invalid blocks. Correspondingly, the LSBs of $\bar{\mathbb{Z}}_{K_l}^l$'s in the first |LM| DCT blocks are fetched and concatenated as LM. Using LM, the remaining DCT blocks (except the invalid blocks) are separated into two types, embeddable blocks and un-embeddable blocks. For the embeddable blocks, the process of extracting secret messages is described by Algorithm 2.

Two illustrations corresponding to the examples in Fig. 6 are given in Fig. 7, to demonstrate the process of extracting the payload bits. The details are described as follows:

1. Example in Fig. 7a

When the RZL pairs in a marked embeddable block are reconstructed, such as [(0/−2), (0/5), (0/−1), (0/18), (0/−1), (1/−1)], N =$2^{\lfloor \log_2(6-1) \rfloor}$=$2^2$ is determined and four RZL pairs [(0/−2), (0/−1), (0/18), (0/−1)] are

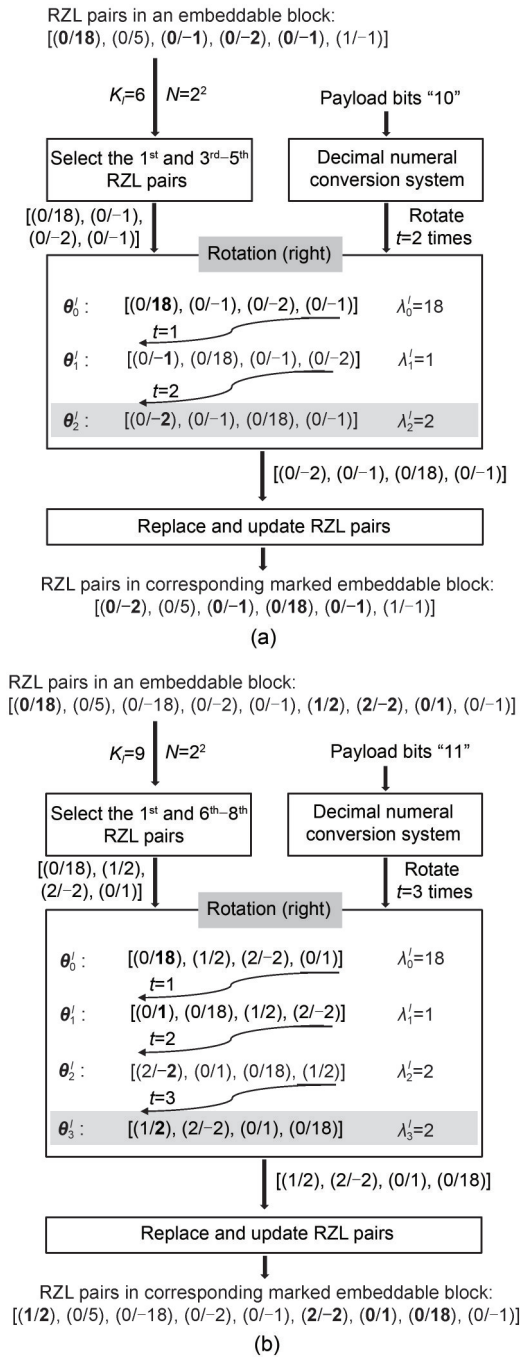

Fig. 6 Two examples of payload embedding: (a) bits "10" into a regular block (N=4); (b) bits "11" into a regular block (N=4)

selected as the initial RZL sequence, i.e., $\bar{\boldsymbol{\theta}}_0^l$. Sequentially, $\bar{\boldsymbol{\theta}}_0^l$ is rotated left $\bar{t}$ times ($\bar{t}$=0, 1, 2, 3) in turn to generate four corresponding RZL sequences, where their features are determined as $\bar{\lambda}_0^l$=2, $\bar{\lambda}_1^l$=1, $\bar{\lambda}_2^l$=18, and $\bar{\lambda}_3^l$=1. It can be apparent that the feature $\bar{\lambda}_2^l$=18 is the unique maximum value among them and the corresponding index is $\bar{t}$=2. Thus, $\log_2 N$=2 payload bits are extracted as $(\bar{t})_2$=$(10)_2$.

Algorithm 2 Extraction of secret messages S

Input: M_{JS} and K_{h}

Output: secret messages S and auxiliary information AI

1 Initialization: payload P=empty
2 Derive pair sequence $\left[\bar{\mathbb{Z}}_1^l, \bar{\mathbb{Z}}_2^l, \cdots, \bar{\mathbb{Z}}_{K_l}^l\right]$ from M_{JS}
3 Identify the invalid blocks and extract LM from LSB of $\bar{\mathbb{Z}}_{K_l}^l$'s in the first |LM| DCT blocks
4 Extract the payload from embeddable blocks. For the l^{th} block:
5 **if** block is embeddable
6 Calculate $N_{\max}=2^{\lfloor \log_2(K_l-1)\rfloor}$, N=$N_{\max}$
7 **while** N≥2
8 Select N RZL pairs $\bar{\boldsymbol{\theta}}^l=\left[\bar{\mathbb{Z}}_1^l, \bar{\mathbb{Z}}_{K_l-N+1}^l, \cdots, \bar{\mathbb{Z}}_{K_l-1}^l\right]$, and set $\bar{\boldsymbol{\theta}}_0^l=\bar{\boldsymbol{\theta}}^l$
9 Rotate left $\bar{\boldsymbol{\theta}}_0^l$ N times to generate N states and obtain $\bar{\boldsymbol{\lambda}}^l=[\bar{\lambda}_0^l, \bar{\lambda}_1^l, \cdots, \bar{\lambda}_{N-1}^l]$
10 **if** $\bar{\boldsymbol{\lambda}}^l=[\bar{\lambda}_0^l, \bar{\lambda}_1^l, \cdots, \bar{\lambda}_{N-1}^l]$ has only one unique maximum value
11 $\bar{t}=\arg\max_a \{\bar{\lambda}_a^l \mid 0\le a\le N-1\}$
12 Convert $\bar{t}$ into binary bits with the length of $\log_2 N$, represented by $(\bar{t})_2$
13 P=P||$(\bar{t})_2$, where "||" represents the concatenation operation
14 **break**
15 **else**
16 N=N/2
17 **end if**
18 **end while**
19 **end if**
20 Repeat step 4 until all embeddable blocks have been processed
21 Fetch AI and ES from P. Decrypt ES to S using K_{h}
22 Output S and AI

2. Example in Fig. 7b

Let us assume that the parsed RZL pairs in an embeddable block are [(1/2), (0/5), (0/−18), (0/−2), (0/−1), (2/−2), (0/1), (0/18), (0/−1)]. In the first round, a sequence consisting of N=2³ RZL pairs, i.e., [(1/2), (0/5), (0/−18), (0/−2), (0/−1), (2/−2), (0/1), (0/18)], is selected and rotated left, resulting in a feature sequence of $\bar{\boldsymbol{\lambda}}^l$=[2, 5, 18, 2, 1, 2, 1, 18]. Apparently, there exist two maximum values, i.e., 18, indicating that the current selected RZL sequence is invalid. In the second

round, N is set to 2^2; thus, the RZL sequence [(1/2), (2/−2), (0/1), (0/18)] is determined. Based on this sequence, four distinct RZL sequences are derived by the operation of rotation left, resulting in feature values of $\bar{\lambda}_0^l$=2, $\bar{\lambda}_1^l$=2, $\bar{\lambda}_2^l$=1, and $\bar{\lambda}_3^l$=18. It can be observed that the index of this unique maximum feature value is $\bar{t}$=3. Therefore, two payload bits "11" are obtained from this embeddable block.

3.6.2 Image recovery

On one hand, the LM and AI can be fetched from M_{JS} in the same way as mentioned in Section 3.6.1. In the following, the LSB of $\mathbb{Z}_{K_l}^l$'s in the first |LM| DCT blocks is recovered by AI. On the other hand, only the embeddable blocks are modified during the data embedding procedure, whereas the un-embeddable blocks and invalid blocks are unchanged. To this end, the embeddable blocks are located according to the LM, and their original encrypted RZL pair sequences can be recovered and updated with the use of the state $\bar{\boldsymbol{\theta}}_{\bar{t}}^l$, where $\bar{t}=\arg\max_a\{\bar{\lambda}_a^l|0\le a\le N-1\}$. Moreover, for ease of understanding, two examples of recovering original encrypted RZL pairs are shown at the bottom of Fig. 7.

If the authorized user holds the encryption key K_{int}, the original plaintext JPEG bitstreams O_{JS} can be recovered by performing the decryption of ACCs and the decryption of DC coefficients. The original plaintext JPEG image can be reconstructed after feeding O_{JS} into the JPEG decoder.

4 Experimental results

This section presents the results for several experiments performed with our system, to demonstrate the effectiveness and superiority of the proposed RDH scheme in encrypted JPEG bitstreams. In our experiments, eight common gray-scale images with a size of 512×512 and images from UCID, BossBase, BOWS-2, and CorelDraw datasets are employed to measure the embedding capacity excluding AI, which is expected

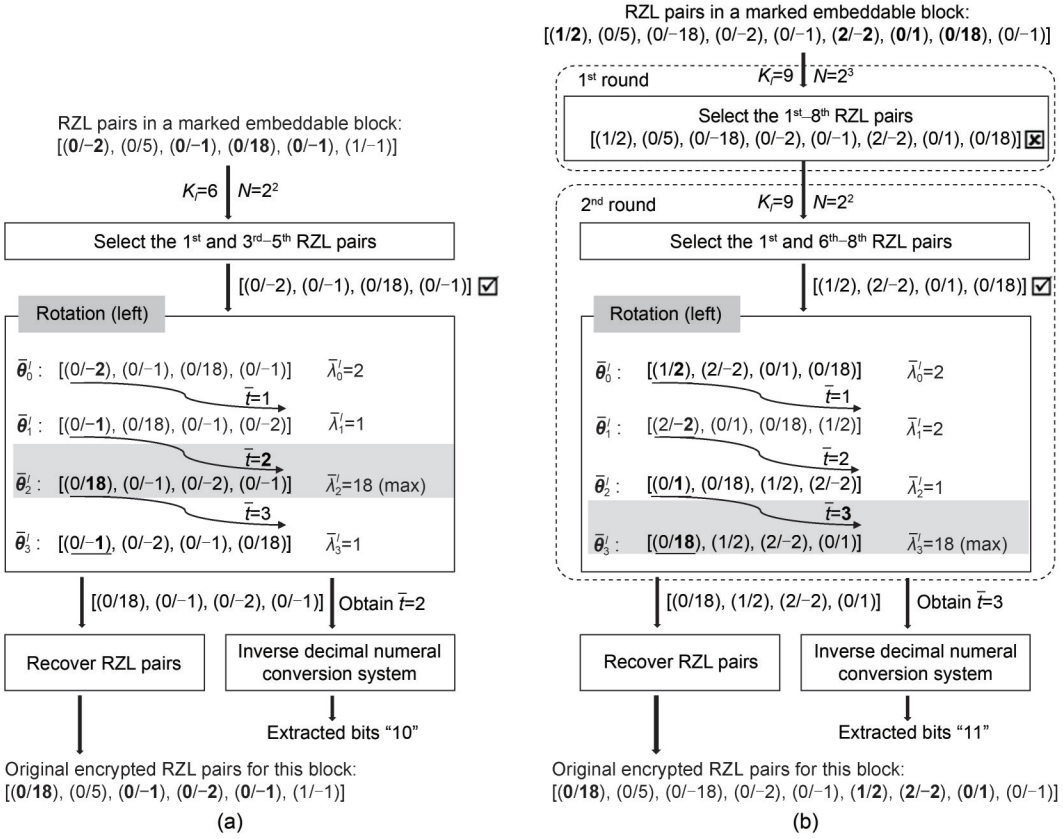

Fig. 7 Two examples of payload extraction: (a) bits "10" from a marked block (N=4); (b) bits "11" from a marked block (N=4)

to be as large as possible to carry the maximum number of secret messages. Here, all images are converted into gray-scale JPEG images, and quality factor (QF), applied to quantize all images in this section, is set to 85, unless specified elsewhere. Moreover, a comparative study is made between the proposed scheme and several recent state-of-the-art schemes, to demonstrate the superior performance of the proposed scheme. All experiments are implemented using MATLAB R2023a on a PC with an Intel® Core™ i5-1035G1 CPU@1.00 GHz 1.19 GHz, 16 GB RAM, and Windows 11 operating system. The security analysis is discussed in the supplementary materials to show that our approach can provide a good security level. It should be pointed out that the initial key and data hiding key should be shared with the receiver over a covert communication channel in advance.

4.1 Performance of embedding capacity

Table 2 presents the results of embedding capacity provided by the proposed scheme when QF is set to 50, 60, 70, 80, and 90. From the results, we can observe that the embedding capacity increases as QF increases. That is because more usable RZL pairs will be generated when a larger QF is employed to process an image block. In particular, when QF is set to 90, the images have an average embedding capacity of 5458.62 bits. It is interesting to note that under the same QF, the embedding capacity of the smoother images is relatively low compared with that of the texture images.

We also experimented on 850 images of size 512×384 from the UCID dataset, 10000 images of size 512×512 from the BossBase dataset, 10000 images of size 512×512 from the BOWS-2 dataset, and 500 images of size 786×512 from the CorelDraw dataset, to further demonstrate the embedding capacity variation tendency. The results are given in Table 3. From the results, we can observe that the variation tendencies of the embedding capacity of our scheme on images from the four datasets are similar; that is, the embedding capacity increases as QF increases. Meanwhile, as expected, the proposed scheme obtains the maximum embedding capacity when QF is set to 90, with average values of 3712.60 bits, 3555.99 bits, 3800.14 bits, and 5718.03 bits, based on UCID, BossBase, BOWS-2, and CorelDraw datasets, respectively. This further confirms that the proposed scheme achieves a satisfactory embedding capacity.

Table 2 Embedding capacity of the proposed scheme on eight common images

Image	Embedding capacity (bit)				
	QF=50	60	70	80	90
Lena	1307	1639	2268	3593	5772
Peppers	1469	1986	2627	4197	5070
Airplane	1512	1637	1971	2921	4393
Couple	1411	1738	2310	3814	5487
Boat	1528	1832	2278	3488	4667
Lake	2490	2663	3293	4885	5886
Aerial	2388	2864	3502	4895	6037
Baboon	2049	2643	3682	5183	6357
Average	1769.25	2125.25	2741.38	4122.00	5458.62

4.2 Performance of file size preservation

The relationships between the file size of the entropy encoded data of the original JPEG bitstreams and that of the marked encrypted JPEG bitstreams under various datasets and QFs are shown in Fig. 8 and the supplementary materials. From the results listed in the 1st and 3rd columns in Fig. 8, we can observe that for a given QF, the solid dots are almost distributed around the diagonal line. Not only that for a given dataset, the solid dots are always distributed around the diagonal area no matter what the QF is. This implies that the file size of the marked encrypted JPEG bitstreams is very close to that of the original JPEG bitstreams. This superior performance is further confirmed by the results plotted in the 2nd and 4th columns,

Table 3 Average embedding capacity of the proposed scheme on images from four datasets

Dataset	Embedding capacity (bit)				
	QF=50	60	70	80	90
UCID (512×384)	1392.58	1626.15	1913.57	2729.10	3712.60
BossBase (512×512)	1157.77	1338.23	1633.21	2518.61	3555.99
BOWS-2 (512×512)	1339.65	1569.36	1887.36	2825.24	3800.14
CorelDraw (768×512)	2152.70	2478.88	2795.10	4111.46	5718.03

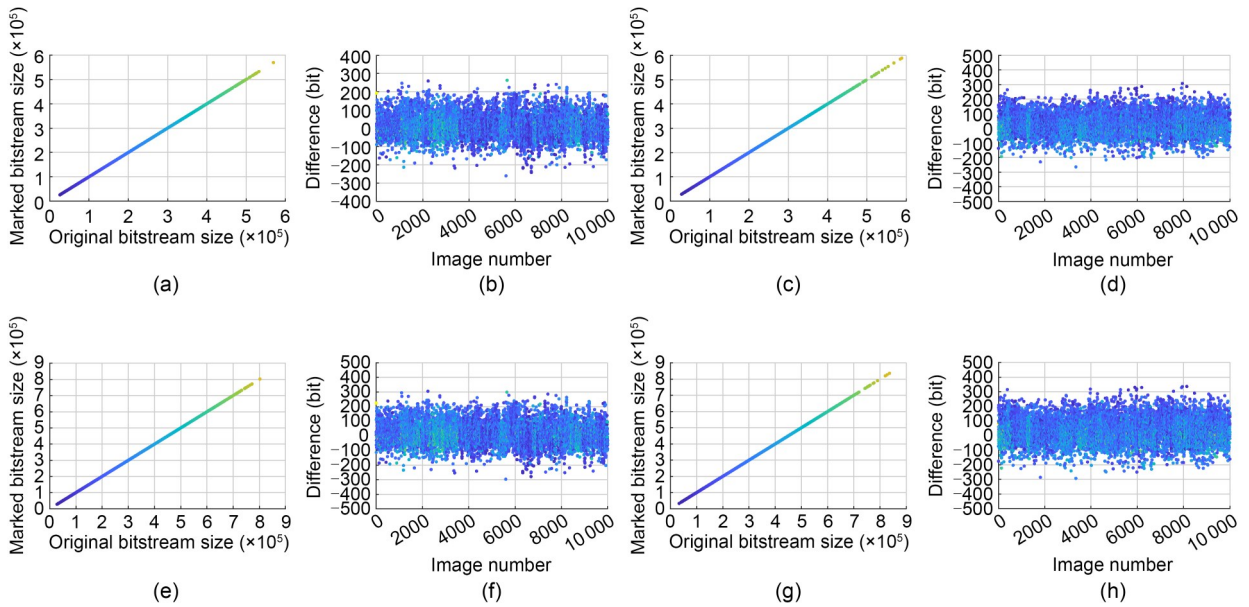

Fig. 8 Illustrations of relationships of file size (bits) between the entropy encoded data of the original JPEG bitstreams and that of the marked encrypted JPEG bitstreams under various datasets and QFs: (a) distribution for BossBase under QF=60; (b) difference for BossBase under QF=60; (c) distribution for BOWS-2 under QF=60; (d) difference for BOWS-2 under QF=60; (e) distribution for BossBase under QF=80; (f) difference for BossBase under QF=80; (g) distribution for BOWS-2 under QF=80; (h) difference for BOWS-2 under QF=80

where most of the differences in the bitstream size of the original JPEG bitstreams and the marked encrypted JPEG bitstreams are located around the zero level for every sub-figure. To summarize, it can be concluded that our proposed RDH scheme in encrypted JPGE bitstreams has the feature of file size preservation.

4.3 Performance analysis and comparison

We compare the performance of the marked encrypted JPEG bitstreams obtained from the proposed scheme with those of other state-of-the-art schemes, with respect to embedding capacity in Fig. 9. From the results, we can observe that the embedding capacities provided by the proposed scheme and the schemes by He JH et al. (2019) amd Hua et al. (2023) are greater than the results provided by the schemes of Qian et al. (2014, 2018) and Chang JC et al. (2017) under every QF. When compared to He JH et al. (2019)'s scheme, where the run-of-zeros of each RZL pair is selected as the key feature, there is a little difference to our method of selecting the levels of non-zero AC coefficients of each RZL as the key feature. Consequently, the embedding capacity of the proposed scheme is somewhat better than that of He JH et al. (2019)'s scheme on images Lena and Couple, whereas the situation is reversed on the images Boat and Baboon. Both approaches have their merits and downsides. As for Hua et al. (2023)'s scheme, since their grouping mechanism boosts the number of embeddable blocks and contributes to increase the candidate permutations, the embedding capacity is ahead.

In addition, we make several comparisons in terms of embedding capacity and file size increment, between our proposed scheme and five state-of-the-art schemes. We use the number of images from three datasets, UCID, BossBase, and CorelDraw. The results are given in Table 4, where partial experimental results are excerpted from the results presented in He JH et al. (2019). Among these, Qian et al. (2019)'s scheme assumes that only 25% of the DCT blocks are inserted into APPn segments. From the results, we can see that on average, the embedding capacity of the proposed scheme is ranked second after Qian et al. (2019)'s scheme and is higher than those of the other schemes (Qian et al., 2014, 2018; Chang JC et al., 2017; He JH et al., 2019). However, for Qian et al. (2019)'s scheme, there exists a serious increment of the file size because of the histogram shifting based data hiding strategy and the re-encoding of 75% of the DC differential coefficients. Thus, the file size of the marked encrypted

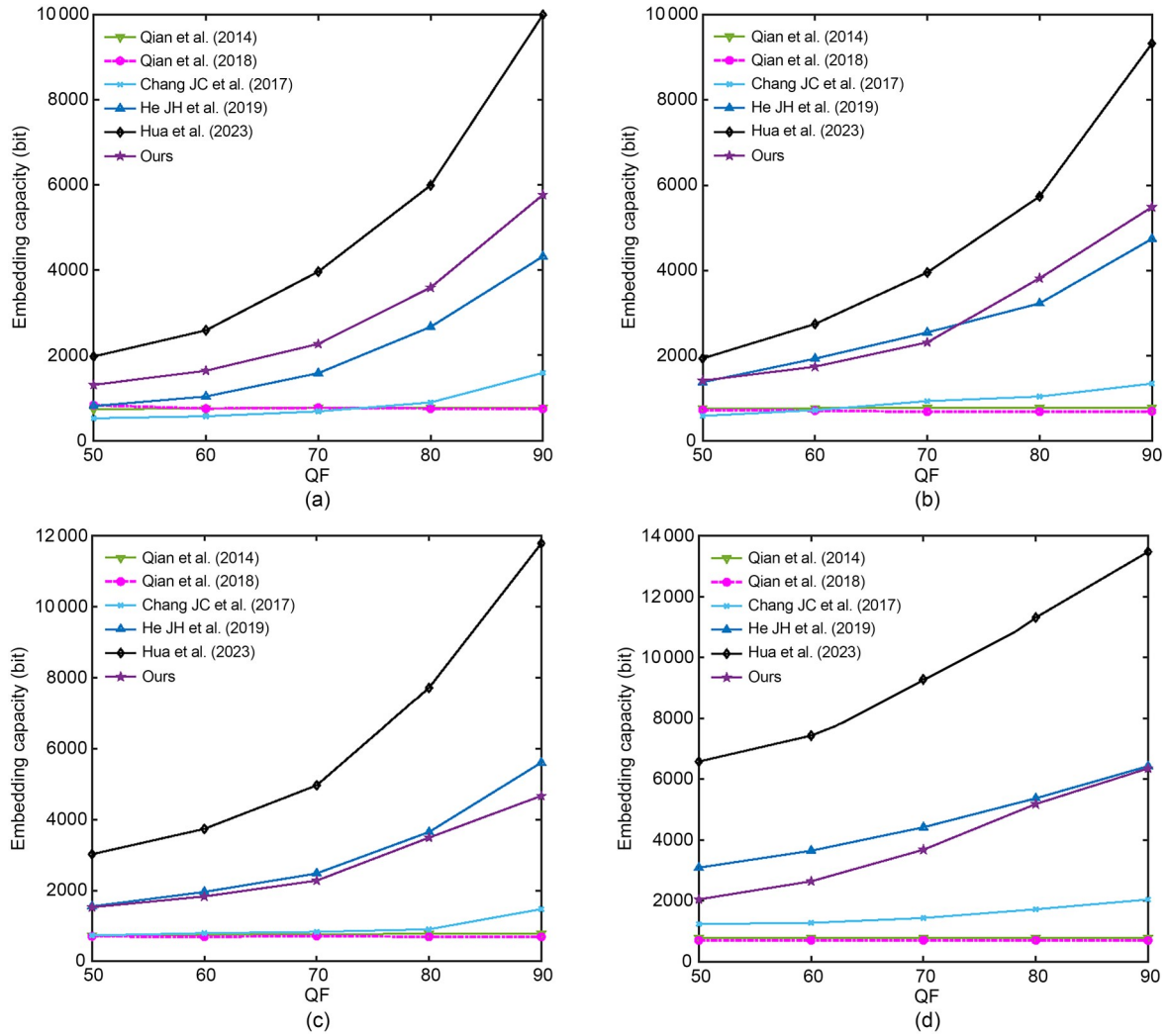

Fig. 9 Embedding capacity of the marked encrypted JPEG bitstreams for different schemes under various QFs: (a) Lena; (b) Couple; (c) Boat; (d) Baboon

Table 4 Embedding capacity and file size increment of the marked encrypted JPEG bitstreams on three datasets when QF=85

Scheme	UCID (512×384)		BossBase (512×512)		CorelDraw (768×512)	
	Embedding capacity (bit)	File size increment (bit)	Embedding capacity (bit)	File size increment (bit)	Embedding capacity (bit)	File size increment (bit)
Qian et al. (2014)	560.59	169.52	729.45	105.52	1122.18	278.12
Qian et al. (2018)	529.80	−1088.12	729.33	−573.76	1060.20	−1555.64
Chang JC et al. (2017)	829.11	154.08	852.33	88.12	1656.90	278.24
Qian et al. (2019)	12228.91	7594.52	11979.10	7692.04	25017.97	14559.24
He JH et al. (2019)	2791.80	120.00	2551.87	74.92	5687.42	165.96
Ours	3146.03	8.38	2937.63	14.11	5713.53	17.52

JPEG bitstreams provided by Qian et al. (2019)'s scheme increases greatly compared to the proposed scheme and other schemes (Qian et al., 2014, 2018; Chang JC et al., 2017; He JH et al., 2019). We also can

observe that the file size increment of our proposed scheme is ranked second after Qian et al. (2018)'s scheme, with values of 8.38 bits, 14.11 bits, and 17.52 bits, based on UCID, BossBase, and CorelDraw, respectively. The analysis and discussions between the payload and peak signal-to-noise ratio (PSNR) and the payload and file size increment under various QFs for related works are further presented in the supplementary materials for comprehensive comparison.

Table 5 summarizes various features for the proposed scheme and related schemes, to briefly clarify the differences among them. Among these, Chang JC et al. (2017)'s scheme requires preserving room for payload embedding in advance and has slight inconformity in format compatibility because of the modification of the mapping relations among the DC differential values. Moreover, the schemes of Qian et al. (2014, 2018, 2019) and Chang JC et al. (2017) need to rewrite the content of JPEG head segment (i.e., JH) due to the requirements of either encrypting the quantization table or filling data into the APPn segment, which is quite different from the schemes of He JH et al. (2019), Hua et al. (2023), Yuan et al. (2023), and ours. The size of an encrypted image in the schemes of Qian et al. (2018, 2019) is necessary to modify due to their specific encryption methods. Moreover, JPEG encryption schemes employed in the schemes of Qian et al. (2014, 2018, 2019) and Chang JC et al. (2017) are not very compatible with the JPEG standard because the overflow of DC coefficients or DC differential coefficients will occur during JPEG decoding. As for Qian et al. (2019)'s scheme and Yuan et al. (2023)'s scheme, there exists a certain file size increment because of their ordered histogram shifting based data hiding. With respect to separability and reversibility, extracting the secret messages error-free in the scheme of Qian et al. (2014) needs the help of their defined block artifact function, and the lossless recovery of the original JPEG image in the scheme of Qian et al. (2018) occurs only if the compression rate of the matrix compression encoding technique is restricted to a lower value. In the schemes of He JH et al. (2019), Hua et al. (2023), and Yuan et al. (2023), the JPEG encryption scheme has considered the format compatibility well, and the number of iterations should be set to be large enough (more than 15) to achieve a relatively satisfactory visual security, resulting in the increment of the computation cost. With respect to embedding capacity, Qian et al. (2019)'s scheme and Yuan et al. (2023)'s scheme are poor at suppressing the file size increment even though they provide a higher embedding capacity. The proposed scheme and other schemes

Table 5 Comparisons of various features for different schemes in JPEG domain

Method	Prereserving room	Need for JPEG header changing	Need for image size changing	File size preservation	Format compatibility	Separability	Reversibility	Capacity	Weakness
Qian et al. (2014)	No	No	Yes	Good	Not ideal	No	Not always	★	DC overflow
Qian et al. (2018)	No	No	No	Good	Not ideal	Yes	Not always	★	DC overflow
Chang JC et al. (2017)	Yes	No	Yes	Good	Not ideal	Yes	Yes	★	DC overflow
Qian et al. (2019)	No	No	No	Increment	Not ideal	Yes	Yes	★★★	DC overflow
He JH et al. (2019)	No	Yes	Yes	Good	Yes	Yes	Yes	★★★	More iterations
Yuan et al. (2023)	No	Yes	Yes	Increment	Yes	Yes	Yes	★★★	More iterations
Hua et al. (2023)	No	Yes	Yes	Good	Yes	Yes	Yes	★★★	More iterations
Ours	No	Yes	Yes	Good	Yes	Yes	Yes	★★	

More ★ means higher embedding capacity

(He JH et al., 2019; Hua et al., 2023) simultaneously perform well in various respects, especially in terms of file size preservation and format compatibility. In addition, we compare the features of different RDH schemes in various encrypted compressed domains, as shown in the supplementary materials, to demonstrate the usability of our proposed scheme.

4.4 Computational complexity analysis and comparison

In Table 6, the computational complexity comparison among different methods is presented. Here, let us assume that the upper bound of the time complexity of an algorithm is Θ, which indicates the worst-case computational complexity. τ is the number of iterations. The detailed analysis of each scheme mentioned in Table 6 is discussed in the supplementary materials to show that the proposed scheme can provide a lower computational cost than other schemes. Meanwhile, we measure the time complexity of the proposed scheme, with the consuming time of 2.29 s. This numerical digit is an average of 886 images, each of which runs nine times in total under varying QFs from 10 to 90. This indicates that the proposed scheme is more efficient and can be deployed in some real-time systems.

5 Conclusions

In this paper, we proposed a novel RDH scheme in encrypted JPEG bitstreams to achieve a satisfactory embedding capacity. Extensive experiments were conducted to confirm the feasibility and effectiveness of our proposed scheme. The main contributions of this paper are summarized as follows:

1. A novel RDH scheme in encrypted JPEG bitstreams toward RZL pairs is put forward. The comparisons of the experimental results showed that the proposed scheme had a superior performance, exceeding the performance of some art-of-the-state schemes in terms of embedding capacity.

2. Meanwhile, the proposed RDH scheme in encrypted JPEG images is quite well compatible with the JPEG standard and effectively suppresses the file size increment.

3. Additionally, our RDH scheme in encrypted JPEG bitstreams has reversibility; that is, the secret messages can be extracted error-free and the original plaintext JPEG image can be recovered losslessly.

In the future, we will try to investigate more applications by using RDH, such as RDH in encrypted JPEG2000 images/H.264 or secret sharing in encrypted dual-compressed images. These may be the interesting subjects. Moreover, we will direct attention towards the development of a robust encryption algorithm to defeat other attacks and improve visual secrecy by introducing other technologies.

Contributors

Yongning GUO, Guodong SU, and Zhiqiang YAO designed the research. Yongning GUO and Guodong SU processed the data and drafted the paper. Zhiqiang YAO helped organize the paper. Yongning GUO, Guodong SU, and Wang ZHOU revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Bai YQ, Jiang GY, Zhu ZJ, et al., 2021. Reversible data hiding scheme for high dynamic range images based on multiple prediction error expansion. *Signal Process Image Commun*, 91:116084. https://doi.org/10.1016/j.image.2020.116084

Table 6 Computational complexity comparison among different schemes

Method	Encryption phase		Embedding phase	Overall
	DC	AC		
He JH et al. (2019)	$(2+\tau)\Theta(L)$	$\Theta(L)+\sum_{l=1}^{L}\Theta(K_l)$	$2N\Theta(L)$	$(3+\tau+2N)\Theta(L)+\sum_{l=1}^{L}\Theta(K_l)$
Yuan et al. (2023)	$(2+\tau)\Theta(L)$	$\Theta(L)+\frac{1}{16}\sum_{l=1}^{L}\Theta(K_l)$	$65\Theta(L)$	$(68+\tau)\Theta(L)+\frac{1}{16}\sum_{l=1}^{L}\Theta(K_l)$
Hua et al. (2023)	$(2+\tau)\Theta(L)$	$\Theta(L)+5\sum_{l=1}^{L}\Theta(K_l)$	$(N^2+2N)\Theta(L)$	$(3+\tau+2N+N^2)\Theta(L)+5\sum_{l=1}^{L}\Theta(K_l)$
Ours	$4\Theta(L)$	$\Theta(L)$	$2N\Theta(L)$	$(5+2N)\Theta(L)$

Chang CC, Li CT, 2019. Algebraic secret sharing using privacy homomorphisms for IoT-based healthcare systems. *Math Biosci Eng*, 16(5):3367-3381. https://doi.org/10.3934/mbe.2019168

Chang CC, Li CT, Shi YQ, 2018. Privacy-aware reversible watermarking in cloud computing environments. *IEEE Access*, 6:70720-70733. https://doi.org/10.1109/ACCESS.2018.2880904

Chang CC, Li CT, Chen KM, 2019. Privacy-preserving reversible information hiding based on arithmetic of quadratic residues. *IEEE Access*, 7:54117-54132. https://doi.org/10.1109/ACCESS.2019.2908924

Chang JC, Lu YZ, Wu HL, 2017. A separable reversible data hiding scheme for encrypted JPEG bitstreams. *Signal Process*, 133:135-143. https://doi.org/10.1016/j.sigpro.2016.11.003

Chuman T, Sirichotedumrong W, Kiya H, 2019. Encryption-then-compression systems using grayscale-based image encryption for JPEG images. *IEEE Trans Inform Forens Secur*, 14(6):1515-1525. https://doi.org/10.1109/TIFS.2018.2881677

He JH, Huang SH, Tang SH, et al., 2018. JPEG image encryption with improved format compatibility and file size preservation. *IEEE Trans Multimed*, 20(10):2645-2658. https://doi.org/10.1109/TMM.2018.2817065

He JH, Chen JX, Luo WQ, et al., 2019. A novel high-capacity reversible data hiding scheme for encrypted JPEG bitstreams. *IEEE Trans Circ Syst Video Technol*, 29(12):3501-3515. https://doi.org/10.1109/TCSVT.2018.2882850

He JH, Chen JX, Tang SH, 2020. Reversible data hiding in JPEG images based on negative influence models. *IEEE Trans Inform Forens Secur*, 15:2121-2133. https://doi.org/10.1109/TIFS.2019.2958758

He WG, Xiong GQ, Weng SW, et al., 2018. Reversible data hiding using multi-pass pixel-value-ordering and pairwise prediction-error expansion. *Inform Sci*, 467:784-799. https://doi.org/10.1016/j.ins.2018.04.088

Hua ZY, Zhou YC, Huang HJ, et al., 2019. Cosine-transform-based chaotic system for image encryption. *Inform Sci*, 480:403-419. https://doi.org/10.1016/j.ins.2018.12.048

Hua ZY, Wang ZY, Zheng YF, et al., 2023. Enabling large-capacity reversible data hiding over encrypted JPEG bitstreams. *IEEE Trans Circ Syst Video Technol*, 33(3):1003-1018. https://doi.org/10.1109/TCSVT.2022.3208030

Huang DL, Wang JJ, 2020. High-capacity reversible data hiding in encrypted image based on specific encryption process. *Signal Process Image Commun*, 80:115632. https://doi.org/10.1016/j.image.2019.115632

Huang FJ, Qu XC, Kim HJ, 2015. Reversible data hiding in JPEG images. *IEEE Trans Circ Syst Video Technol*, 26(9):1610-1621. https://doi.org/10.1109/TCSVT.2015.2473235

Interantional Telecommunication Union, 1992. Information Technology—Digital Compression and Coding of Continuous-Tone Still Images—Requirements and Guidelines. Recommendation T.81, ITU.

Li XL, Li B, Yang B, et al., 2013. General framework to histogram-shifting-based reversible data hiding. *IEEE Trans Image Process*, 22(6):2181-2191. https://doi.org/10.1109/TIP.2013.2246179

Ma KD, Zhang WM, Zhao XF, et al., 2013. Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Trans Inform Forens Secur*, 8(3):553-562. https://doi.org/10.1109/TIFS.2013.2248725

Mohammadi A, Nakhkash M, Akhaee MA, 2020. A high-capacity reversible data hiding in encrypted images employing local difference predictor. *IEEE Trans Circ Syst Video Technol*, 30(8):2366-2376. https://doi.org/10.1109/TCSVT.2020.2990952

Ni ZC, Shi YQ, Ansari N, et al., 2006. Reversible data hiding. *IEEE Trans Circ Syst Video Technol*, 16(3):354-362. https://doi.org/10.1109/TCSVT.2006.869964

Ou B, Li XL, Zhao Y, et al., 2013. Pairwise prediction-error expansion for efficient reversible data hiding. *IEEE Trans Image Process*, 22(12):5010-5021. https://doi.org/10.1109/TIP.2013.2281422

Pareek NK, Patidar V, Sud KK, 2006. Image encryption using chaotic logistic map. *Image Vis Comput*, 24(9):926-934. https://doi.org/10.1016/j.imavis.2006.02.021

Peng F, Zhao Y, Zhang X, et al., 2020. Reversible data hiding based on RSBEMD coding and adaptive multi-segment left and right histogram shifting. *Signal Process Image Commun*, 81:115715. https://doi.org/10.1016/j.image.2019.115715

Puteaux P, Puech W, 2018. An efficient MSB prediction-based method for high-capacity reversible data hiding in encrypted images. *IEEE Trans Inform Forens Secur*, 13(7):1670-1681. https://doi.org/10.1109/TIFS.2018.2799381

Puteaux P, Wang ZC, Zhang XP, et al., 2021. Hierarchical high capacity data hiding in JPEG crypto-compressed images. Proc 28th European Signal Process Conf, p.725-729. https://doi.org/10.23919/Eusipco47968.2020.9287376

Qi KL, Zhang MQ, Di FQ, et al., 2023. High capacity reversible data hiding in encrypted images based on adaptive quadtree partitioning and MSB prediction. *Front Inform Technol Electron Eng*, 24(8):1156-1168. https://doi.org/10.1631/FITEE.2200501

Qian ZX, Zhang XP, Wang SZ, 2014. Reversible data hiding in encrypted JPEG bitstream. *IEEE Trans Multimed*, 16(5):1486-1491. https://doi.org/10.1109/TMM.2014.2316154

Qian ZX, Zhou H, Zhang XP, et al., 2018. Separable reversible data hiding in encrypted JPEG bitstreams. *IEEE Trans Depend Secur Comput*, 15(6):1055-1067. https://doi.org/10.1109/TDSC.2016.2634161

Qian ZX, Xu HS, Luo XY, et al., 2019. New framework of reversible data hiding in encrypted JPEG bitstreams. *IEEE Trans Circ Syst Video Technol*, 29(2):351-362. https://doi.org/10.1109/TCSVT.2018.2797897

Qin C, Hu YC, 2016. Reversible data hiding in VQ index table with lossless coding and adaptive switching mechanism. *Signal Process*, 129:48-55. https://doi.org/10.1016/j.sigpro.2016.05.032

Qin C, Zhang W, Cao F, et al., 2018. Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection. *Signal Process*, 153:109-122. https://doi.org/10.1016/j.sigpro.2018.07.008

Qiu YQ, Qian ZX, Zeng HQ, et al., 2020. Reversible data hiding in encrypted images using adaptive reversible integer

transformation. *Signal Process*, 167:107288. https://doi.org/10.1016/j.sigpro.2019.107288

Rabbani M, Joshi R, 2002. An overview of the JPEG 2000 still image compression standard. *Signal Process Image Commun*, 17(1):3-48. https://doi.org/10.1016/S0923-5965(01)00024-8

Su GD, Chang CC, 2023. Toward high-capacity crypto-domain reversible data hiding with Huffman-based lossless image coding. *Vis Comput*, 39(10):4623-4638. https://doi.org/10.1007/s00371-022-02613-z

Su GD, Chang CC, Lin CC, 2019. A square lattice oriented reversible information hiding scheme with reversibility and adaptivity for dual images. *J Vis Commun Image R*, 64: 102618. https://doi.org/10.1016/j.jvcir.2019.102618

Su GD, Chang CC, Lin CC, et al., 2023. Towards property-preserving JPEG encryption with structured permutation and adaptive group differentiation. *Vis Comput*, 40:6421-6447. https://doi.org/10.1007/s00371-023-03174-5

Tian J, 2003. Reversible data embedding using a difference expansion. *IEEE Trans Circ Syst Video Technol*, 13(8): 890-896. https://doi.org/10.1109/TCSVT.2003.815962

Wang WQ, Ye JY, Wang TQ, et al., 2017. Reversible data hiding scheme based on significant-bit-difference expansion. *IET Image Process*, 11(11):1002-1014. https://doi.org/10.1049/iet-ipr.2017.0151

Wu HT, Cheung YM, Yang ZY, et al., 2019. A high-capacity reversible data hiding method for homomorphic encrypted images. *J Visual Commun Image Represent*, 62:87-96. https://doi.org/10.1016/j.jvcir.2019.04.015

Xiao D, Wang Y, Xiang T, et al., 2017. High-payload completely reversible data hiding in encrypted images by an interpolation technique. *Front Inform Technol Electron Eng*, 18(11): 1732-1743. https://doi.org/10.1631/FITEE.1601067

Xie EY, Li CQ, Yu SM, et al., 2017. On the cryptanalysis of Fridrich's chaotic image encryption scheme. *Signal Process*, 132:150-154. https://doi.org/10.1016/j.sigpro.2016.10.002

Yi S, Zhou YC, Hua ZY, 2018. Reversible data hiding in encrypted images using adaptive block-level prediction-error expansion. *Signal Process Image Commun*, 64:78-88. https://doi.org/10.1016/j.image.2018.03.001

Ying QC, Qian ZX, Zhang XP, et al., 2019. Reversible data hiding with image enhancement using histogram shifting. *IEEE Access*, 7:46506-46521. https://doi.org/10.1109/ACCESS.2019.2909560

Yuan Y, He HJ, Chen F, et al., 2023. Reversible data hiding in encrypted JPEG image with changing the number of AC codes. *Multimed Tools Appl*, 82(28):43649-43669. https://doi.org/10.1007/s11042-023-14614-8

Zhang LY, Liu S, Pareschi F, et al., 2018. On the security of a class of diffusion mechanisms for image encryption. *IEEE Trans Cybern*, 48(4):1163-1175. https://doi.org/10.1109/tcyb.2017.2682561

Zhang WM, Hu XC, Li XL, et al., 2013. Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression. *IEEE Trans Image Process*, 22(7):2775-2785. https://doi.org/10.1109/TIP.2013.2257814

Zhang XP, 2012. Separable reversible data hiding in encrypted image. *IEEE Trans Inform Forens Secur*, 7(2):826-832. https://doi.org/10.1109/TIFS.2011.2176120

Zhou LN, Lu ZG, You WK, et al., 2023. Reversible data hiding using a transformer predictor and an adaptive embedding strategy. *Front Inform Technol Electron Eng*, 24(8): 1143-1155. https://doi.org/10.1631/FITEE.2300041

List of supplementary materials

1 Security analysis and comparison

2 Performance of file size preservation under various QFs

3 Comparison of PSNR among different schemes

4 Comparison of payload and file size increment under various QFs

5 Comparison of features for different RDH schemes in the encrypted compressed domain

6 Computational complexity analysis and comparison

Table S1 Key space for four typical JPEG images

Table S2 Statistical data for the four typical JPEG images

Table S3 Security comparison between the proposed scheme and other schemes

Table S4 Average payload and file size increment of our scheme and Hua et al. (2023)'s scheme on images from the BossBase dataset

Table S5 Comparison of features for different RDH schemes in encrypted compressed domain

Fig. S1 Visual effect for Lena and Baboon images when QF=85

Fig. S2 Sketch attack analysis for the proposed scheme

Fig. S3 Illustration of the relationships of file size between the entropy encoded data of the original JPEG bitstreams and those of the marked encrypted JPEG bitstreams under various datasets and QFs

Fig. S4 PSNR of the marked approximate JPEG images under various payloads when QF=85