Hindawi
Security and Communication Networks
Volume 2022, Article ID 1227926, 16 pages
https://doi.org/10.1155/2022/1227926

Research Article

Separable Reversible Data Hiding in Encrypted VQ-Encoded Images

Fang Cao,[1,2] **Yujie Fu**,[3] **Heng Yao**,[3] **Mian Zou**,[4] **Jian Li**,[5] **and Chuan Qin**[3]

[1]*College of Information Engineering, Shanghai Maritime University, Shanghai 200135, China*
[2]*Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, Guilin 541004, China*
[3]*School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China*
[4]*School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China*
[5]*School of Cyber Security, Qilu University of Technology (Shandong Academy of Sciences), Shandong Provincial Key Laboratory of Computer Networks, Jinan 250353, China*

Correspondence should be addressed to Jian Li; ljian20@gmail.com

Received 21 December 2021; Accepted 16 February 2022; Published 23 April 2022

Academic Editor: Beijing Chen

In this paper, a reversible data-hiding scheme in encrypted, vector quantization (VQ) encoded images is proposed. During image encryption, VQ-encoded image, including codebook and index table, is encrypted by content owner with stream-cipher and permutation to protect the privacy of image contents. As for additional-data embedding, a baseline method is first proposed and its corresponding optimized method is then given. By grouping one high-occurrence index with one or multiple low-occurrence indices, a series of index groups are constructed. Thus, by modifying the high-occurrence index to the corresponding index within the same group according to the current to-be-embedded bits, data embedding can be realized. The optimal hiding capacity is obtained by optimizing the coefficient vector for different types of index groups. Separable operations of data extraction, image decryption, and recovery can be achieved on the receiver side based on the availability of the encryption and data-hiding keys. Experimental results show that our scheme can achieve high hiding capacity and satisfactory directly decrypted image quality and guarantee security and reversibility simultaneously.

1. Introduction

With the rapid development of digital communication and signal processing, a large amount of multimedia data, such as image, video, and audio, are transmitted on networks. However, secure management for multimedia data with privacy protection is an inevitable issue and also is one meaningful research topic. Reversible data-hiding (RDH) is an emerging technique which has greatly attracted researchers' interests in recent years [1–4]. As for the RDH technique, data hider can embed additional data into cover image reversibly, which means the original cover image can be completely recovered after extracting the embedded data. Some representative RDH schemes, such as difference expansion (DE) [1], histogram shifting (HS) [2], and prediction-error expansion (PEE) [3], have been proposed in the past few years. In addition to the RDH schemes for gray scale images, there are a lot of RDH schemes developed for color images [5], compressed images [6], and halftone images [7].

Vector quantization (VQ) is an effective image encoding method, which can be utilized for image compression [8]. During the process of VQ encoding, the original uncompressed image $\mathbf{I}_o$ is first divided into a series of nonoverlapping blocks. For each image block, Euclidean distances between the block with the L code words in the trained codebook $\mathbf{C}$ are calculated, and the index of the code word with the minimum Euclidean distance is recorded into the index table $\mathbf{T}$ as the encoded result for the current block. During VQ decoding, according to the indices in the index table **T**, all image blocks can be easily decoded as the corresponding code words in the codebook **C** to form the VQ-decoded image **I**. Figure 1 illustrates an example of VQ-encoded image, in which a gray scale image with the size of $M \times N$ is compressed to an index table sized $M/n \times N/n$, where $n \times n$ is the size of divided blocks. Each value in the index table, corresponding to one $n \times n$ block, can be represented with $\log_2 L$ bits. Thus, for the whole image, the compression ratio can be calculated as $8 \times n^2/\log_2 L$. Generally speaking, a codebook with more code words, i.e., larger L, can lead to better visual quality of VQ-decoded image.

In recent years, a number of RDH schemes have been developed for VQ-encoded images in the plaintext form [9–16]. Chang et al. proposed a RDH scheme in VQ-encoded images based on a de-clustering strategy [9], in which two de-clustering methods were used with the minimum-spanning-tree and a short-spanning-path. Lee et al. modified VQ-encoded images by the side-matched VQ (SMVQ) technique to form a transformed image, and exploited the distribution of this transformed image to achieve high hiding capacity and low bit rate [10]. Kieu and Ramroach utilized the joint neighboring coding method to realize reversible steganographic scheme for VQ indices [12], in which the differences between the current index; the left, upper, and top-left neighboring indices; and their combinations were used to hide additional bits. In Ref. [15], two RDH schemes for VQ-encoded images were proposed based on switching-tree coding and dynamic-tree coding. These two schemes performed data embedding by choosing one of the possible index encoding ways when multiple ways were available to encode the index, and the outputted codes can be decoded to original VQ index table with the conventional decoder. Pan and Wang proposed a RDH scheme for two-stage VQ-encoded image based on search-order coding (SOC) in Ref. [16]. SOC can employ the correlation of indices to obtain better compression ratio, thus, the combination of SOC and data hiding in this scheme can achieve both high performances for compression ratio and hiding capacity.

Due to the current prosperity of cloud storing and computing, a vast amount of personal data are stored and processed on the cloud to alleviate computation burden on user clients [17, 18]. But, in order to protect user privacy, it is better to first encrypt user data before uploading onto cloud. Thereby, for the convenience of data management and retrieval, RDH in encrypted images (RDHEI) has attracted extensive interest in the field of multimedia security. According to when the space for accommodating additional data was created, i.e., before or after image encryption, embedding mechanisms of most RDHEI schemes can be categorized into two types: vacating room after encryption (VRAE) [19–28] and reserving room before encryption (RRBE) [29–34]. In addition, some researchers introduced homomorphic encryption (HE) into RDHEI [35–38], which can realize the operations of data embedding directly in encrypted domain. A brief review of the related works on RDHEI is given in Section 2.

In this work, we focus on RDH in encrypted, VQ-encoded image. An encryption method for VQ-encoded image is first designed for the codebook and the index table, respectively. Before conducting additional-data embedding in the encrypted index table, all VQ indices are sorted according to their occurrence numbers. A baseline method of data embedding is proposed based on constructing index groups for one high-occurrence index and one low-occurrence index each time, and then we improve the baseline method through generalized index grouping for multiple low-occurrence indices. By modifying the high-occurrence index to the corresponding index within the same group according to the current to-be-embedded bits, additional-data embedding can be achieved, and the optimal hiding capacity is obtained by optimizing coefficient vector for different types of index groups. Separable operations of data extraction, image decryption, and recovery can be realized on the receiver side based on the availability of the encryption and data-hiding keys. The proposed scheme can achieve satisfactory performances of hiding capacity and directly decrypted image quality and guarantee security and reversibility simultaneously.

The remaining parts of the paper are organized as follows. Section 2 gives a brief review of related works about RDHEI. Section 3 introduces the baseline of the proposed scheme, including image encryption, additional-data embedding, data extraction, and image recovery. Section 4 gives performance optimization for additional-data embedding procedure of the baseline method in Section 3, which consists of generalized index grouping, multiple-bits embedding, and hiding capacity optimization. Section 5 presents experimental results and analysis. Conclusions are drawn in Section 6.

2. Related Works

An effective RDHEI framework can be described as: the content-owner encrypts the original image with encryption key and then sends the encrypted image to the data hider; the data hider embeds additional data into the encrypted image with data-hiding key to produce the marked, encrypted image; and the authorized receiver implements data extraction, image decryption, and image recovery on the marked, encrypted image according to encryption key and data-hiding key. In the following, three main categories of RDHEI schemes are briefly reviewed.

2.1. VRAE-Based Schemes. In Ref. [19], the encrypted image with stream cipher was segmented into a number of non-overlapping blocks, and by flipping the three LSBs of different parts of pixels, one bit of additional data can be embedded into each block. The receiver can achieve data extraction and image recovery through estimation with a fluctuation function. Hong et al. improved the order of data extraction and block recovery and introduced a side-match strategy to increase the accuracy of the extracted data and recovered image [20]. Liao and Shu utilized the absolute mean difference of neighboring pixels to measure the

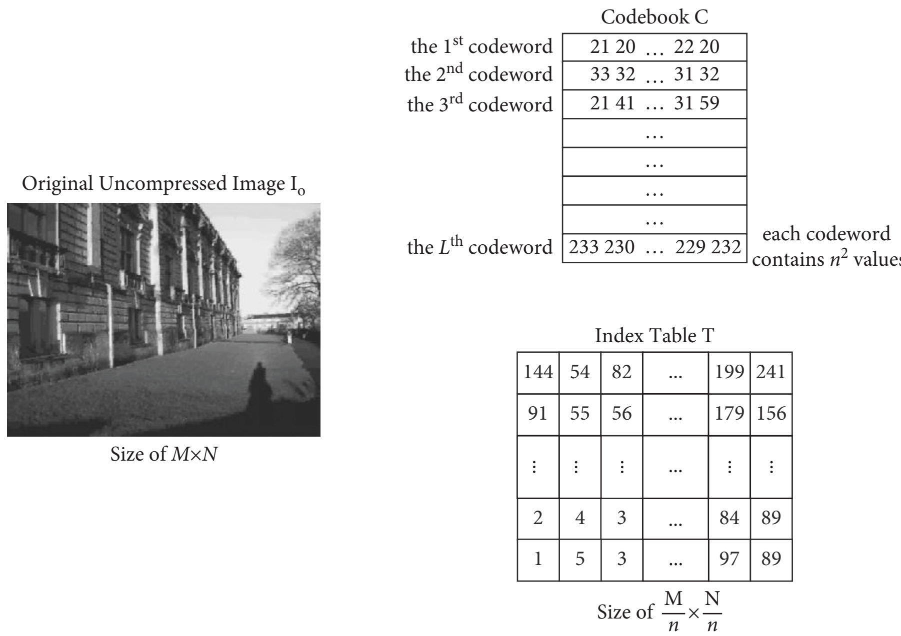


Figure 1: An example of VQ-encoded image.

recovery accuracy of image blocks after decryption [21]. Different from Refs. [19–21] that data extraction must be conducted after image decryption, a separable RDHEI scheme was proposed in Ref. [22], which means that the operations of image decryption and data extraction can be realized on the receiver side independently. A public key modulation mechanism was employed in Ref. [23] to achieve data embedding without accessing the secret encryption key. In addition, a powerful two-class SVM classifier was presented to differentiate the encrypted and nonencrypted patches, leading to recovering the embedded data and original image correctly. Huang et al. proposed to encrypt the original image in a blockwise manner [24], which can retain the correlation within the pixels of each encrypted block. Then, traditional RDH methods in plaintext images can be used in encrypted image for data hiding. In Ref. [25], a RDHEI scheme with an adaptive encoding strategy was presented, which adaptively compressed the MSB layers of embeddable blocks according to occurrence frequency of MSB and then embedded additional data together with reversed Huffman codewords and auxiliary data. Yi and Zhou first presented a parametric binary tree labeling (PBTL) algorithm to label pixels in two different types, and then, they proposed a PBTL-RDHEI scheme in encrypted images, which can achieve data embedding by pixel labeling and bit replacement effectively [28].

2.2. RRBE-Based Schemes. In order to avoid the errors on data extraction or image recovery, Ma et al. proposed a scheme by reserving room before encryption with a traditional RDH method in Ref. [29], which can acquire the complete reversibility. In Ref. [30], some pixels in the original plaintext image were first predicted before encryption, thus, additional data can then be embedded in the prediction errors. A benchmark encryption algorithm was applied to the rest pixels and a specific encryption algorithm was designed to encrypt prediction errors. Cao et al. considered that an image patch can be linearly represented by some atoms in an over-complete dictionary through sparse coding [31], and the residual errors can be encoded and self-embedded in the original image. Thereby, a large extra room can be created before image encryption, and the data hider can embed more additional data into the encrypted image based on this strategy of patch-level sparse representation. Puteaux et al. proposed a new reversible method with most significant bit (MSB) prediction [32], which can achieve a high hiding capacity. During the preprocessing, a location map was produced by detecting prediction errors. Through MSB substitution, additional data can be embedded and the embedding rate was close to 1 bpp. Yin et al. proposed a RDHEI scheme based on multi-MSB prediction and Huffman coding [33]. Before image encryption with a stream cipher, multi-MSB of each pixel was predicted and marked with Huffman coding in the original image as the preprocessing. Thus, additional data can be embedded into the encrypted image through multi-MSB substitution.

2.3. HE-Based Schemes. Chen et al. proposed a RDH scheme for encrypted signals based on Paillier public key encryption, and applied it on digital images [35], in which each pixel value was divided into two parts encrypted, respectively. Then, two encrypted LSBs of each pixel pair were modified to hide one bit with the help of homomorphism. In Wu et al.'s scheme [36], each unit in the original image was segmented into three components with energy transfer equation, and each component was encrypted by Paillier homomorphic encryption. The data hider can embed additional bits into the encrypted image by using the properties of Paillier homomorphism. A separable RDHEI scheme based on additive homomorphism and pixel value ordering (PVO) was given in Ref. [37]. Additive homomorphism applied in this scheme can guarantee that the performance of embedding rate for PVO in an encrypted domain can approximate to that in plaintext domain without involving data expansion. In Ref. [38], Xiang and Luo proposed to form mirroring ciphertext groups (MCGs) by replacing encrypted host pixels with encrypted reference pixels in the same group. In an MCG, the reference ciphertext pixel remained unchanged as a reference while the data hider can embed additional data into the LSBs of host encrypted pixels with homomorphic multiplication.

The abovementioned RDHEI schemes mainly focused on the encrypted, uncompressed gray scale image. In addition, some schemes have also been designed for other kinds of cover data, such as JPEG-encoded image [39–41], palette image [42], 2D vector graphic [43], and 3D mesh model [44], in the encrypted domain. However, to the best of our knowledge, there are few reported works about RDHEI of VQ-encoded images currently.

3. Baseline of the Proposed Scheme

Figure 2 presents the framework of the proposed scheme for RDH in encrypted VQ-encoded images. As shown in Figure 2(a), on the content-owner side, with encryption key $K_e = \{K_e^{(1)}, K_e^{(2)}\}$, encryption for VQ-encoded image can be divided into two steps: codebook encryption and index table encryption, respectively. Then, after receiving the encrypted, VQ-encoded image, through index grouping and data-hiding key K_h, additional data can be embedded on the data-hider side, see Figure 2(b). On the receiver side, we can extract additional data and restore the VQ-encoded image. It can be seen from Figure 2(c) that additional data can be extracted with data-hiding key K_h; receiver can obtain a decrypted image which is similar to the original image with encryption key K_e; when the receiver has both encryption key K_e and data-hiding key K_h, the embedded data can be successfully extracted and the VQ-encoded image can also be perfectly recovered. Details of our baseline scheme are introduced as follows.

3.1. VQ-Encoded Image Encryption. As we know, a VQ-encoded image consists of one codebook $\mathbf{C}$ and an index table $\mathbf{T}$. Hence, in order to guarantee the security, VQ-encoded image encryption can be divided into two parts, i.e., codebook encryption and index table encryption.

Suppose VQ codebook $\mathbf{C}$ contains L code words, and in each code word, there are n^2 decimal values. Denote $P_{i,j}$ as the j^{th} value of the i^{th} code word in the codebook $\mathbf{C}$, where $i = 1, 2, \ldots, L$, $j = 1, 2, \ldots, n^2$, and the value of $P_{i,j}$ can be represented as eight binary bits:

$$P_{i,j,k} = \left\lfloor \frac{P_{i,j}}{2^{k-1}} \right\rfloor \bmod 2, \quad k = 1, 2, \ldots, 8, \tag{1}$$

where $P_{i,j,k}$ denotes the k^{th} bit of $P_{i,j}$. A sequence of pseudo-random bits $S_{i,j,k}$ $(i = 1, 2, \ldots, L, j = 1, 2, \ldots, n^2, k = 1, 2, \ldots, 8)$ is generated with encryption key $K_e^{(1)}$. The operation of bitwise exclusive-or (XOR) is performed on all L code words for codebook encryption:

$$\begin{aligned} P_{i,j,k}^{(e)} &= P_{i,j,k} \oplus S_{i,j,k}, \\ P_{i,j}^{(e)} &= \sum_{k=1}^{8} P_{i,j,k}^{(e)} \cdot 2^{k-1}, \end{aligned} \tag{2}$$

where ${P_{i,j}}^{(e)}$ denotes the j^{th} encrypted value in the i^{th} code word after stream-cipher encryption. After all code words in the codebook $\mathbf{C}$ are encrypted, the encrypted codebook $\mathbf{C}_e$ is obtained.

As for the index table $\mathbf{T}$ sized $M/n \times N/n$, all the index values in $\mathbf{T}$ are permuted with the encryption key $K_e^{(2)}$.

$$\mathbf{T}_{\mathrm{e}} = perm\left(\mathbf{T}, K_{\mathrm{e}}^{(2)}\right), \tag{3}$$

where perm(·) denotes the permutation function, and $\mathbf{T}_e$ is the encrypted index table. The security can be guaranteed because the codebook $\mathbf{C}$ is encrypted by the stream cipher while permuting the index table $\mathbf{T}$. The key space of index table permutation can be calculated as $(M/n \times N/n)!$ As for an original uncompressed image sized 512×512 $(M = N = 512)$, when block size is chosen as 4×4 $(n = 4)$, the whole key space of index table permutation is: $[(512/4) \times (512/4)]! = 16384!$ The codebook encryption based on stream cipher can be considered to further strengthen the security of encryption, even when the permutation key $K_e^{(2)}$ is leaked or cracked. After the VQ-encoded image encryption for codebook and index table, $\mathbf{C}_{\mathrm{e}}$ and $\mathbf{T}_{\mathrm{e}}$ are transmitted to the data-hider side together for additional-data embedding.

3.2. Additional-Data Embedding. In our scheme, after receiving $\mathbf{C}_e$ and $\mathbf{T}_e$, data hider first counts the occurrence numbers of VQ indices in the encrypted index table $\mathbf{T}_e$. Initially, the occurrence numbers of indices corresponding to all L code words in the encrypted codebook $\mathbf{C}_e$ are set as zero. When a VQ index is scanned in $\mathbf{T}_e$, its occurrence number is increased by one. That is to say, for the VQ index k, its occurrence number γ_k $(k = 1, 2, \ldots, L)$ can be calculated as:

$$\gamma_k = \sum_{x=1}^{M/n} \sum_{y=1}^{N/n} \phi\left(k, T_{x,y}\right), \tag{4}$$

where $T_{x,\,y}$ denotes the index value at coordinate (x, y) in the index table $\mathbf{T}_e$, and $\varphi(\cdot)$ is a counting function returning 1 or 0. When the current VQ index $T_{x,\,y}$ is equal to k, φ returns 1; otherwise, φ returns 0. Generally, occurrence numbers of VQ indices in the index table are not uniform for a natural image. Figure 3 shows the distribution of occurrence numbers of VQ indices $(L = 128)$ for image *Lena*, in which X axis and Y axis denote the index values and their corresponding occurrence numbers, respectively. We can observe from Figure 3 that some VQ indices occur frequently while some VQ indices are not used at all.

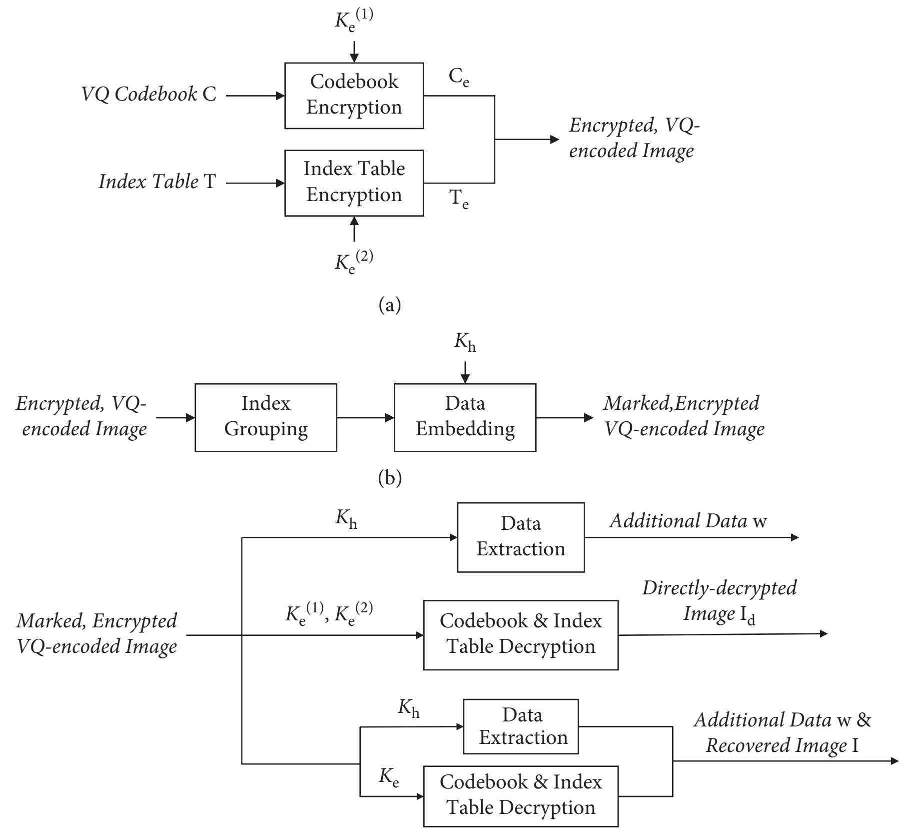


Figure 2: Framework of the proposed scheme. (a) VQ-encoded image encryption, (b) additional-data embedding, (c) data extraction and image recovery.

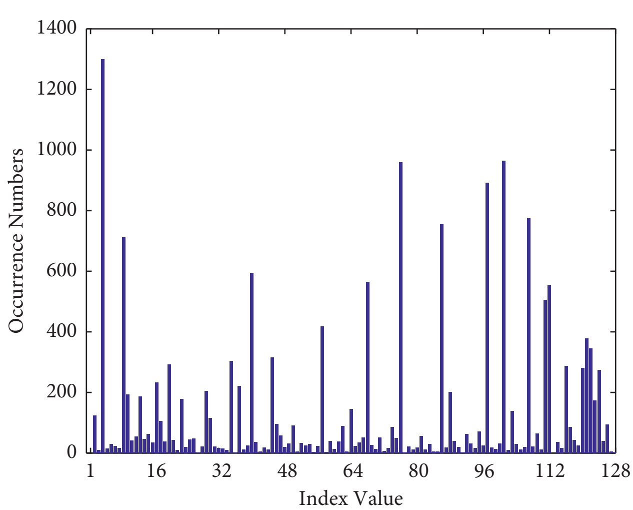


Figure 3: Histogram of VQ index values for image *Lena* ($L = 128$).

After scanning the whole index table $\mathbf{T}_e$, the L different kinds of VQ indices are sorted according to their corresponding occurrence numbers γ_k ($k = 1, 2, \ldots, L$) in the descending order. In the proposed scheme, VQ indices with higher and lower occurrence numbers are utilized to achieve additional-data embedding. Note that the distribution of VQ indices is not changed before and after VQ-encoded image encryption in our scheme. The procedure of additional-data embedding includes two stages: (1) index grouping and (2) data embedding, which are described in detail as follows.

3.2.1. A Index Grouping. Denote the sorted L VQ indices as $c_1, c_2, \ldots, c_L$, and their corresponding occurrence numbers are $\gamma_1, \gamma_2, \ldots, \gamma_L$. We define the index set $\{c_\alpha, c_{\alpha+1}, \ldots, c_L\}$ with lower occurrence numbers as $\mathbf{\Phi}$, where α is a threshold satisfying $\gamma_\alpha \le \sigma$, and σ is a pre-determined parameter, which is discussed in Section 5. The relationship between α and σ can be represented as:

$$\alpha = \arg\min_i \gamma_i \le \sigma. \quad (5)$$

In addition to the VQ indices in $\mathbf{\Phi}$, the VQ indices with higher occurrence numbers are selected as another set $\mathbf{\Theta} = \{c_1, c_2, \ldots, c_\beta\}$, and their corresponding index occurrence numbers are $\gamma_1, \gamma_2, \ldots, \gamma_\beta$, where β is set to $L - \alpha + 1$. The remaining indices are formed as the set $\mathbf{\Omega} = \{c_{\beta+1}, c_{\beta+2}, \ldots, c_{\alpha-1}\}$.

In the following, VQ indices from the two sets $\mathbf{\Phi}$ and $\mathbf{\Theta}$ are exploited to construct β index groups through an iterative strategy. In detail, the β index groups are emptied initially. Then, the index with the highest occurrence number, denoted as $c_g^{(j)}$, in the current set $\mathbf{\Theta}$ and the index with the lowest occurrence number, denoted as $c_s^{(j)}$, in the current set $\mathbf{\Phi}$ are selected to form one index group $\{c_g^{(j)}, c_s^{(j)}\}$, $j = 1, 2, \ldots, \beta$, and the two indices, $c_g^{(j)}$ and $c_s^{(j)}$, are removed from $\mathbf{\Theta}$ and $\mathbf{\Phi}$, respectively. According to the above way, β index groups can be constructed iteratively until the two sets $\mathbf{\Theta}$ and $\mathbf{\Phi}$ become empty.

3.2.2. Data Embedding. In order to guarantee the reversibility of original VQ-encoded image on the receiver side, side information should be recorded for the indices in $\mathbf{\Phi}$ whose occurrence numbers are not zero, i.e., $c_i \in \mathbf{\Phi}$ and $\gamma_i \neq 0$, $i \in \{\alpha, \alpha+1, \ldots, L\}$. In detail, for the jth index group $\{c_g^{(j)}, c_s^{(j)}\}$, $j = 1, 2, \ldots, \beta$, we first utilize $\log_2(MN/n^2)$ bits to sequentially represent the occurrence number of the index $c_s^{(j)}$ in $\mathbf{T}_e$; if the occurrence number of the index $c_s^{(j)}$ in $\mathbf{T}_e$, i.e., γ_i corresponding to $c_i \in \mathbf{\Phi}$, does not equal 0, we should further utilize $\gamma_i \cdot \log_2(MN/n^2)$ bits to record the position information of the γ_i indices in $\mathbf{T}_e$. Thus, the length of side information is:

$$\rho = \left(\beta + \sum_{i=\alpha}^{L} \gamma_i\right) \cdot \log_2\left(\frac{MN}{n^2}\right). \tag{6}$$

We compress the side information by run-length coding and concatenate the compressed side information with the additional data $\mathbf{w}$ to be embedded together as w' after scrambling with the data-hiding key K_h. During data embedding, for each index group $\{c_g^{(j)}, c_s^{(j)}\}$, if the occurrence number of the index $c_s^{(j)}$ in $\mathbf{T}_e$ is not equal to 0, data hider should replace all the index values $c_s^{(j)}$ in $\mathbf{T}_e$ with the indices c_m that can be randomly selected from the set $\mathbf{\Omega}$, and the modified index table is denoted as $\mathbf{T}'_e$. Then, each VQ index in $\mathbf{T}'_e$ that is equal to the index $c_g^{(j)}$ with higher occurrence number in one of the β index groups, i.e., $\{c_g^{(j)}, c_s^{(j)}\}$, can be embedded with one binary bit. In detail, data hider scans the VQ indices in $\mathbf{T}'_e$ with the raster-scanning order, and if the current scanning index $T_{x,y}$ is equal to $c_g^{(j)}$ in the jth index group ($j = 1, 2, \ldots, \beta$), one binary bit w_i from w' can be embedded by:

$$T'_{x,y} = \begin{cases} c_g^{(j)}, & \text{if } w_i = 0, \\ c_s^{(j)}, & \text{if } w_i = 1, \end{cases} \tag{7}$$

where $T'_{x,y}$ denotes the marked VQ index. In other words, for the current scanning index $T_{x,y}$ belonging to the set $\mathbf{\Theta}$, if the to-be-embedded bit is 0, $T_{x,y}$ remains unchanged, otherwise, $T_{x,y}$ is changed to its corresponding index with lower occurrence number in the same index group.

After all VQ indices, belonging to $\mathbf{\Theta}$, in $\mathbf{T}'_e$ finish the above procedure, we can obtain a marked, encrypted index table $\mathbf{T}_{ew}$. Then, $\mathbf{T}_{ew}$ and $\mathbf{C}_e$ are transmitted to the receiver for data extraction and image recovery. Note that the β index groups should also be sent to the receiver as auxiliary data $\mathfrak{R}$.

3.3. Data Extraction and Image Recovery. When the receiver obtains the marked, encrypted index table $\mathbf{T}_{ew}$, the encrypted codebook $\mathbf{C}_e$ and the auxiliary data $\mathfrak{R}$, data extraction and image recovery can be conducted. There are three scenarios: (1) if the receiver only has the data-hiding key K_h, the additional data $\mathbf{w}$ can be extracted correctly; (2) if the receiver only has the encryption key K_e, a directly decrypted index table $\mathbf{T}_d$, which is similar to the original index table $\mathbf{T}$ can be obtained; and (3) if the receiver has both K_e and K_h, additional data $\mathbf{w}$ and original index table $\mathbf{T}$ can both be recovered with no error. Details are presented as follows.

3.3.1. Data Extraction. First, the two index sets $\mathbf{\Theta}$ and $\mathbf{\Phi}$ corresponding to higher and lower occurrence numbers can be easily obtained based on the auxiliary data $\mathfrak{R}$ representing the β index groups. Then, during scanning the index table $\mathbf{T}_{ew}$ in the raster-scanning order, according to the current scanning index $T'_{x,y}$ belonging to $\mathbf{\Theta}$ or $\mathbf{\Phi}$, the embedded bit w'_i can be extracted sequentially, see equation (8).

$$w'_i = \begin{cases} 0, & \text{if } T'_{x,y} \in \mathbf{\Theta}, \\ 1, & \text{if } T'_{x,y} \in \mathbf{\Phi}, \\ \text{no data extracted}, & \text{other wise}. \end{cases} \tag{8}$$

Concatenating all extracted bits w'_i, the embedded data w' can be obtained correctly. Then, the receiver inversely scrambles w' through the data-hiding key K_h, and parses the ρ-bits side information from w', thus, the remaining part is the extracted additional data $\mathbf{w}$.

3.3.2. Image Decryption. If the receiver only has the encryption key $K_e = \{K_e^{(1)}, K_e^{(2)}\}$, he/she can first generate the sequence of pseudo-random bits $S_{i,j,k}$ ($i = 1, 2, \ldots, L$, $j = 1, 2, \ldots, n^2$, $k = 1, 2, \ldots, 8$) by $K_e^{(1)}$, which is the same with the one on the content-owner side. Through decrypting based on XOR operation, the decrypted codebook $\mathbf{C}_d$ can be obtained, which is exactly the same as the original codebook $\mathbf{C}$.

On the other hand, the receiver scans the marked, encrypted index table $\mathbf{T}_{ew}$, and if the current scanning index $T'_{x,y}$ is equal to $c_s^{(j)}$ in one of the β index groups ($j = 1, 2, \ldots, \beta$), $T'_{x,y}$ is modified as the corresponding $c_g^{(j)}$ in the same group. After all indices in $\mathbf{T}_{ew}$ are performed, a new index table $\mathbf{T}'_e$ can be produced. Then, through decrypting $\mathbf{T}'_e$ based on permutation with $K_e^{(2)}$, a directly decrypted index table $\mathbf{T}_d$, which is similar to the original index table $\mathbf{T}$, can be obtained. If required, a directly decrypted image $\mathbf{I}_d$ can also be acquired through decoding the index table $\mathbf{T}_d$ by the VQ codebook $\mathbf{C}$.

As we know, the side information records the numbers and the positions for the indices in $\mathbf{\Phi}$ with nonzero occurrence numbers, and these indices are replaced by c_m randomly selected in $\mathbf{\Omega}$ during data embedding. Therefore, due to the unavailability of the data-hiding key K_h, the side information cannot be parsed from w', which means these recorded indices cannot be recovered from c_m in $\mathbf{\Omega}$ to $c_s^{(j)}$ in $\mathbf{\Phi}$. In other words, without K_h, $\mathbf{T}'_e$ cannot be recovered to $\mathbf{T}_e$ perfectly, and $\mathbf{T}_d$ is not exactly the same with $\mathbf{T}$.

3.3.3. Image Recovery. If both encryption key K_e and data-hiding key K_h are available, the receiver can obtain the index table $\mathbf{T}'_e$ and can also parse the ρ-bits side information and the additional data $\mathbf{w}$ from the extracted data w' with K_h. According to the parsed side information, the receiver can know the detailed numbers and positions for the indices in $\mathbf{\Phi}$ with nonzero occurrence numbers, which are replaced by c_m randomly selected in $\mathbf{\Omega}$ during data embedding. Thus, through scanning the index table $\mathbf{T}_e'$, the indices at the positions indicated in the side information can be restored from c_m to the corresponding index $c_s^{(j)}$ (whose occurrence number is not zero) in the index group$\{c_g^{(j)}, c_s^{(j)}\}$. As a result, the index table $\mathbf{T}_e'$ is recovered to $\mathbf{T}_e$ perfectly, and after decrypting $\mathbf{T}_e$ based on permutation with $K_e^{(2)}$, the original index table $\mathbf{T}$ can be recovered reversibly. As described previously, the original codebook $\mathbf{C}$ can be obtained through decrypting $\mathbf{C}_d$ with $K_e^{(1)}$ based on XOR operation. Therefore, the additional data $\mathbf{w}$, original index table $\mathbf{T}$, and original codebook $\mathbf{C}$ can all be acquired when both K_e and K_h are available. If required, the original VQ-decoded image $\mathbf{I}$ can also be acquired through decoding $\mathbf{T}$ by $\mathbf{C}$.

4. Performance Optimization

The proposed scheme described in Section 3 can be considered as the baseline, which can be further optimized on the performance of hiding capacity during additional-data embedding. The optimization mainly focuses on adaptively adjusting the number of indices with lower occurrence numbers in index groups. Note that encryption operation on the content-owner side is unchanged. Details are given as follows.

4.1. Generalized Index Grouping. In the optimized scheme, during index grouping on the data-hider side, we define that, in each group, the number of indices with higher occurrence numbers is fixed as 1, while the number of indices with lower occurrence numbers should be $2^{\nu}-1$, where ν is a variable integer satisfying:

$$\nu \in \{1, 2, \ldots, \lfloor \log_2 (L-\alpha+1) \rfloor\}. \tag{9}$$

Thus, one index $c_g^{(j)}$ with higher occurrence numbers selected from $\mathbf{\Theta} = \{c_1, c_2, \ldots, c_\beta\}$ and $2^{\nu}-1$ indices, $c_s^{(j,1)}, c_s^{(j,2)}, \ldots, c_s^{(j,2^{\nu}-1)}$, with lower occurrence numbers selected from $\mathbf{\Phi} = \{c_\alpha, c_{\alpha+1}, \ldots, c_L\}$ can be constructed as one index group, i.e., $\{c_g^{(j)}, c_s^{(j,1)}, c_s^{(j,2)}, \ldots, c_s^{(j,2^{\nu}-1))}\}$. Note that if ν is a constant equaling 1 for all groups, the optimized scheme is just the baseline proposed in Section 3. In addition, different from the baseline scheme in Section 3, in the optimized scheme, the number β of the indices with higher occurrence numbers in $\mathbf{\Theta} = \{c_1, c_2, \ldots, c_\beta\}$ may not be equal to the number $(L-\alpha+1)$ of the indices with lower occurrence numbers in $\mathbf{\Phi} = \{c_\alpha, c_{\alpha+1}, \ldots, c_L\}$.

We consider the index groups including the same number $2^{\nu}-1$ of lower occurrence indices as the same (i.e., the νth) type of index groups, and a coefficient μ_ν is defined to represent the number of index groups belonging to the νth type, $\nu = 1, 2, \ldots, \log_2(L-\alpha+1)$. A coefficient vector $\boldsymbol{\mu}$ can be given for different types of index groups, see equation (10). Table 1 lists the detailed information for different types of index groups.

Table 1: Details of different types of existing index groups.

Group type	$\nu = \log_2(L-\alpha+1)$	...	$\nu = 2$	$\nu = 1$
The number of c_g in group	1	...	1	1
The number of c_s in group	$2^{\nu}-1$	...	2^2-1	2^1-1
Group coefficient	μ_ν	...	μ_2	μ_1
Embedding ability (bits)	$\log_2(L-\alpha+1)$	...	2	1

$$\boldsymbol{\mu} = \left[\mu_1, \mu_2, \ldots, \mu_{\lfloor \log_2 (L-\alpha+1) \rfloor}\right]. \tag{10}$$

The generalized index grouping should satisfy the following two relationships:

$$L-\alpha+1 = \sum_{\nu=1}^{\lfloor \log_2 (L-\alpha+1) \rfloor} \mu_\nu \cdot (2^{\nu}-1), \tag{11}$$

$$\beta = \sum_{\nu=1}^{\lfloor \log_2 (L-\alpha+1) \rfloor} \mu_\nu. \tag{12}$$

Equation (11) implies that $\mu_\nu \cdot (2^{\nu}-1)$ represents the number of indices from $\mathbf{\Phi}$ belonging to the νth type of index group, and in all $\log_2(L-\alpha+1)$ types of index groups, the total number of indices from $\mathbf{\Phi}$ should be equal to $L-\alpha+1$. On the other hand, Equation (12) guarantees that each index group has one index from $\mathbf{\Theta}$.

For intuitive description, we present an example of generalized index grouping in Figure 4. Figure 4(a) shows the sorted indices and their corresponding code words ($L = 128$), which are sorted in the descending order according to occurrence numbers of indices within index table $\mathbf{T}_e$. Here, we set the parameter σ in equation (4) to 1, thereby, α can be derived as 116. Thus, we can obtain $128 - 116 + 1 = 13$ indices with lower occurrence numbers in $\mathbf{\Phi}$, i.e., $\{c_{116}, c_{117}, \ldots, c_{128}\}$. Then, as shown in Figure 4(b), after generalized index grouping, there are three types of index groups (including five index groups totally), corresponding to $\nu = 3$ ($\mu_3 = 1$), $\nu = 2$ ($\mu_2 = 1$), $\nu = 1$ ($\mu_1 = 3$), respectively. The value of β can also be obtained as $1 + 1 + 3 = 5$ through equation (12), which means that five indices with higher occurrence numbers are included in $\mathbf{\Theta}$, i.e., $\{c_1, c_2, \ldots, c_5\}$. In detail, $\{c_1, c_{122}, c_{123}, c_{124}, c_{125}, c_{126}, c_{127}, c_{128}\}$ belongs to the third type of index group ($\nu = 3$, $\mu_3 = 1$); $\{c_2, c_{119}, c_{120}, c_{121}\}$ belongs to the second type of index group ($\nu = 2$, $\mu_2 = 1$); $\{c_3, c_{118}\}$, $\{c_4, c_{117}\}$, and $\{c_5, c_{116}\}$ belong to the first type of index groups ($\nu = 1$, $\mu_1 = 3$). The coefficient vector $\boldsymbol{\mu}$ is equal to $[1, 1, 3]$. Table 2 summarizes the index grouping information for the example in Figure 4.

4.2. Multiple-Bits Embedding. Similar with the baseline scheme described in Section 3, we also need to sequentially record the occurrence numbers of the indices belonging to $\mathbf{\Phi}$ and their positions (if existing) in $\mathbf{T}_e$ as the ρ-bits side information, see equation (6). Then, side information can be compressed through run-length coding and be concatenated with the additional data $\mathbf{w}$ as w' after scrambling with the data-hiding key K_h. During data embedding, for any index in $\mathbf{\Phi}$ whose occurrence number is not equal to 0, data hider should replace this index value in $\mathbf{T}_e$ with the index c_m that

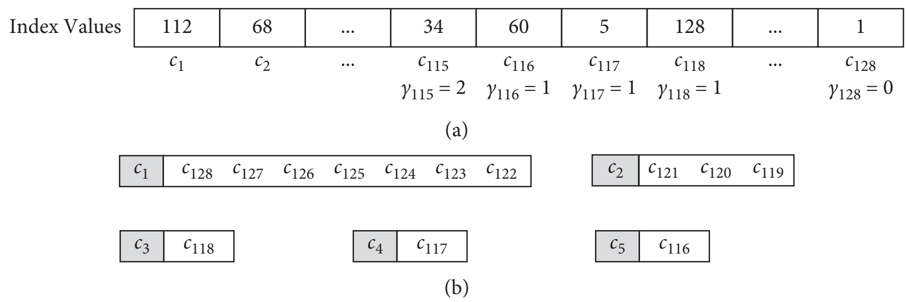


Figure 4: An example of generalized index grouping. (a) The sorted indices, (b) A result of index grouping.

Table 2: Index grouping information for the example in Figure 4.

Group Type	$\nu=3$	$\nu=2$	$\nu=1$
The number of c_g in group	1	1	1
The number of c_s in group	7	3	1
Group coefficient	1	1	3
Embedding ability (bits)	3	2	1

can be randomly selected from the set $\mathbf{\Omega}$, and the modified index table is also denoted as $\mathbf{T}'_e$. Then, each VQ index in $\mathbf{T}'_e$ that is equal to the index $c_g^{(j)}$ with higher occurrence number in the generalized index group, i.e., $\{c_g^{(j)}, c_s^{(j,1)}, c_s^{(j,2)}, \ldots, c_s^{(j,2^\nu-1))}\}$, can be embedded with ν binary bits ($j=1, 2, \ldots, \beta$). In detail, data hider scans the VQ indices in $\mathbf{T}'_e$ with the raster-scanning order, and if the current scanning index $T_{x,y}$ is equal to $c_g^{(j)}$ in the jth index group, multiple bits, $\{w_i, w_{i+1}, \ldots, w_{i+v-1}\}$, from $\mathbf{w}'$ can be embedded by:

$$T'_{x,y} = \begin{cases} c_g^{(j)}, & \text{if } \boldsymbol{\tau}_i = 0, \\ c_s^{(j,1)}, & \text{if } \boldsymbol{\tau}_i = 1, \\ \cdots & \cdots \\ c_s^{(j,2^\nu-1)}, & \text{if } \boldsymbol{\tau}_i = 2^\nu - 1, \end{cases} \tag{13}$$

where $\boldsymbol{\tau}_i$ denotes the decimal value of the current ν bits $\{w_i, w_{i+1}, \ldots, w_{i+v-1}\}$ for embedding, and $T'_{x,y}$ denotes the marked VQ index embedded with the ν bits. Equation (13) means that, for the current scanning index $T_{x,y}$ belonging to the set $\mathbf{\Theta}$, if the decimal value $\boldsymbol{\tau}_i$ of the current to-be-embedded ν bits is 0, $T_{x,y}$ remains unchanged, otherwise, $T_{x,y}$ is changed to the $\boldsymbol{\tau}_i$th corresponding index with a lower occurrence number in the same index group. After all VQ indices, belonging to $\mathbf{\Theta}$, in $\mathbf{T}'_e$ are scanned and performed with above operations orderly, the procedure of multiple-bits embedding is finished and the marked, encrypted index table $\mathbf{T}_{\text{ew}}$ can be acquired.

Continuing the example in Figure 4, since $\{c_1, c_{122}, c_{123}, c_{124}, c_{125}, c_{126}, c_{127}, c_{128}\}$ belongs to the third type of index group ($\nu=3$), three binary bits can be embedded when the current scanning index is c_1; since $\{c_2, c_{119}, c_{120}, c_{121}\}$ belongs to the second type of index group ($\nu=2$), two binary bits can be embedded when the current scanning index is c_2; since $\{c_3, c_{118}\}$, $\{c_4, c_{117}\}$ and $\{c_5, c_{116}\}$ belong to the first type of index group ($\nu=1$), one binary bit can be embedded when the current scanning index is c_3, c_4, or c_5. It can be inferred that, after data embedding, the occurrence numbers of c_1, c_2, c_3, c_4 and c_5 in the index table are decreased because a portion of them are changed to the indices with lower occurrence number in their corresponding index groups.

Obviously, if index groups are determined, the hiding capacity ζ of the optimized scheme can be calculated. The occurrence numbers for the VQ indices with higher occurrence numbers, $\{c_1, c_2, \ldots, c_\beta\}$, in $\mathbf{\Theta}$ are denoted as $\mathbf{f}=[\gamma_1, \gamma_2, \ldots, \gamma_\beta]$. According to the coefficient vector $\boldsymbol{\mu}$ in equation (10), the occurrence number vector $\mathbf{f}$ can be transformed to a $1\times\log_2(L-\alpha+1)$ vector $\boldsymbol{\eta}=[\eta_1, \eta_2, \ldots, \eta_{\log2(L-\alpha+1)}]$:

$$\eta_\nu = \begin{cases} 0, & \mu_\nu = 0, \\ \sum_{i=\kappa_\nu+1}^{\kappa_\nu+\mu_\nu} \gamma_i, & \mu_\nu \neq 0, \end{cases} \quad \nu = 1, 2, \ldots, \lfloor \log_2(L-\alpha+1) \rfloor, \tag{14}$$

$$\kappa_\nu = \begin{cases} 0, & \nu = \lfloor \log_2(L-\alpha+1) \rfloor, \\ \sum_{j=\nu+1}^{\lfloor \log_2(L-\alpha+1) \rfloor} \mu_j, & \nu < \lfloor \log_2(L-\alpha+1) \rfloor, \end{cases} \tag{15}$$

where η_ν denotes the number of indices with higher occurrence numbers belonging to the νth type of index group, $\nu=1, 2, \ldots, \log_2(L-\alpha+1)$. Note that, the lengths of the two vectors $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ are equal. An example of transform procedure from $\mathbf{f}$ to $\boldsymbol{\eta}$ is given in Figure 5. Assume that the occurrence number vector is f = [1299, 964, 960, 891, 775, 755, 711] and the coefficient vector is $\mu=[1, 2, 3, 0, 1]$, respectively. With the assistance of $\boldsymbol{\mu}$, the vector $\boldsymbol{\eta}$ can be obtained as [711, 775 + 755, 964 + 960+891, 0, 1299] = [711, 1530, 2815, 0, 1299].

Therefore, based on the above descriptions, the hiding capacity ζ of the proposed scheme can be obtained as:

$$\zeta = \sum_{\nu=1}^{\lfloor \log_2(L-\alpha+1) \rfloor} \nu \cdot \eta_\nu, \tag{16}$$

The operations on the receiver side, including data extraction, image decryption, and recovery after receiving the marked, encrypted index table $\mathbf{T}_{\text{ew}}$, the encrypted codebook $\mathbf{C}_{\text{e}}$, and the auxiliary data $\mathfrak{R}$ of the β index group, have minor differences with those described in Section 3. As for data extraction, $\mathbf{T}_{\text{ew}}$ is first scanned in the raster-scanning order, and if the current scanning index $T'_{x,y}$ is equal to one of the indices in an index group, i.e., $\{c_g^{(j)}, c_s^{(j,1)}, c_s^{(j,2)}, \ldots, c_s^{(j,2^\nu-1))}\}$, $j=1, 2, \ldots, \beta$, ν binary bits can be

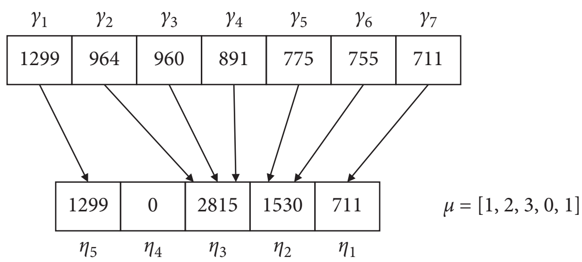


Figure 5: An example of transform procedure from $\mathbf{f}$ to $\boldsymbol{\eta}$.

extracted according to equation (13), and then, the side information and additional data can be parsed with K_h. As for image decryption, $T'_{x,y}$ is modified as corresponding to the high-occurrence index $c_g^{(j)}$ in the same group, thus, $\mathbf{T}_{ew}$ is changed as $\mathbf{T}'_e$ after all indices in $\mathbf{T}_{ew}$ are scanned. Then, the original VQ codebook $\mathbf{C}$ and the directly decrypted index table $\mathbf{T}_d$ are produced through decrypting $\mathbf{C}_e$ and $\mathbf{T}'_e$ with $K_e^{(1)}$ and $K_e^{(2)}$, respectively. The directly decrypted image $\mathbf{I}_d$ can be obtained by decoding $\mathbf{T}_d$ by $\mathbf{C}$. As for image recovery, the index table $\mathbf{T}'_e$ should be first recovered to $\mathbf{T}_e$ with the assistance of side information. As we know, side information sequentially records the occurrence numbers and the positions (if existing) for the $L-\alpha+1$ indices belonging to $\boldsymbol{\Phi}$ with lower occurrence numbers in $\mathbf{T}_e$, among which those indices with nonzero occurrence numbers are replaced by c_m randomly selected in $\boldsymbol{\Omega}$ during data embedding. Therefore, with the parsed side information, the receiver can sequentially restore c_m back to the corresponding index with nonzero, lower occurrence numbers in $\{c_\alpha, c_{\alpha+1}, \ldots, c_L\}$. As a result, $\mathbf{T}'_e$ can be recovered to $\mathbf{T}_e$ perfectly, and after decrypting $\mathbf{T}_e$, original index table $\mathbf{T}$ can be obtained. Finally, original VQ-decoded image $\mathbf{I}$ can also be acquired through decoding $\mathbf{T}$ by $\mathbf{C}$.

4.3. Hiding Capacity Optimization. It should be noticed that there possibly exist multiple coefficient vectors $\boldsymbol{\mu}$ that can satisfy the two relationships in equations (11) and (12), and different coefficient vectors $\boldsymbol{\mu}$ may lead to different hiding capacities. Therefore, by finding the optimal coefficient vector, hiding capacity of the proposed scheme can be further optimized.

Suppose that there are λ different coefficient vectors, $\boldsymbol{\mu}^{(1)}$, $\boldsymbol{\mu}^{(2)}, \ldots, \boldsymbol{\mu}^{(\lambda)}$, satisfying equations (11) and (12), which can be represented by a matrix $\mathbf{U}$ sized $\lambda \times \log_2(L-\alpha+1)$:

$$\mathbf{U} = \begin{bmatrix} \boldsymbol{\mu}^{(1)} \\ \boldsymbol{\mu}^{(2)} \\ \vdots \\ \boldsymbol{\mu}^{(\lambda)} \end{bmatrix} = \begin{bmatrix} \mu_1^{(1)} & \mu_2^{(1)} & \cdots & \mu_{\lfloor \log_2(L-\alpha+1)\rfloor}^{(1)} \\ \mu_1^{(2)} & \mu_2^{(2)} & \cdots & \mu_{\lfloor \log_2(L-\alpha+1)\rfloor}^{(2)} \\ \vdots & \vdots & \cdots & \vdots \\ \mu_1^{(\lambda)} & \mu_2^{(\lambda)} & \cdots & \mu_{\lfloor \log_2(L-\alpha+1)\rfloor}^{(\lambda)} \end{bmatrix}, \quad (17)$$

where $\boldsymbol{\mu}^{(i)} = [\mu_1^{(i)}, \mu_2^{(i)}, \ldots, \mu_{\lfloor \log_2(L-\alpha+1)\rfloor}^{(i)}]$, $i = 1, 2, \ldots, \lambda$. According to equations (14) and (15), we can know that each row in the matrix $\mathbf{U}$, i.e., $\boldsymbol{\mu}^{(i)}$, corresponds to a vector $\boldsymbol{\eta}^{(i)}$ based on the occurrence number vector $\mathbf{f} = [\gamma_1, \gamma_2, \ldots, \gamma_\beta]$. Thus, the λ vectors, $\boldsymbol{\eta}^{(1)}, \boldsymbol{\eta}^{(2)}, \ldots, \boldsymbol{\eta}^{(\lambda)}$, can form a matrix $\boldsymbol{\Gamma}$ that has the same size with $\mathbf{U}$:

$$\Gamma = \begin{bmatrix} \boldsymbol{\eta}^{(1)} \\ \boldsymbol{\eta}^{(2)} \\ \vdots \\ \boldsymbol{\eta}^{(\lambda)} \end{bmatrix} = \begin{bmatrix} \eta_1^{(1)} & \eta_2^{(1)} & \cdots & \eta_{\lfloor \log_2(L-\alpha+1)\rfloor}^{(1)} \\ \eta_1^{(2)} & \eta_2^{(2)} & \cdots & \eta_{\lfloor \log_2(L-\alpha+1)\rfloor}^{(2)} \\ \vdots & \vdots & \cdots & \vdots \\ \eta_1^{(\lambda)} & \eta_2^{(\lambda)} & \cdots & \eta_{\lfloor \log_2(L-\alpha+1)\rfloor}^{(\lambda)} \end{bmatrix}, \quad (18)$$

where $\boldsymbol{\eta}^{(i)} = [\eta_1^{(i)}, \eta_2^{(i)}, \ldots, \eta_{\lfloor \log_2(L-\alpha+1)\rfloor}^{(i)}]$, $i = 1, 2, \ldots, \lambda$. Then, based on equation (16), we can obtain λ values, $\zeta^{(1)}, \zeta^{(2)}, \ldots, \zeta^{(\lambda)}$, of the hiding capacity:

$$\zeta^{(i)} = \mathbf{r} \cdot \left(\boldsymbol{\eta}^{(i)}\right)^{\mathrm{T}}, \quad (19)$$

where $\mathbf{r}$ denotes the row vector $[1, 2, \ldots, \log_2(L-\alpha+1)]$, and $\zeta^{(i)}$ is the hiding capacity corresponding to the coefficient vector $\boldsymbol{\mu}^{(i)}$ for index groups, $i = 1, 2, \ldots, \lambda$. With equation (20), the optimal coefficient vector can be found as $\boldsymbol{\mu}^{(i*)}$ by dynamic programming, which means that the optimal result of generalized index grouping is determined. Finally, the largest hiding capacity of the proposed scheme after optimization can be acquired as $\zeta^{(i*)}$.

$$i^* = \arg\max_i \zeta^{(i)}, \quad \text{subject to } i \in \{1, 2, \ldots, \lambda\}. \quad (20)$$

5. Experimental Results and Analysis

In order to demonstrate the effectiveness and superiority of our scheme, experiments were conducted on a large number of VQ-encoded images, and the environment of our experiments was based on a personal computer with a 3.20 GHz Intel i5 processor, 4.00 GB memory, Windows 10 operating system, and Matlab R2016a. In the following, results of the proposed scheme, including the reversibility, hiding capacity ζ, and visual quality of directly decrypted image $\mathbf{I}_d$, are first given. Then, the influences of parameter σ on the performances are analyzed. Finally, comparisons with state-of-the-art schemes are discussed.

5.1. Results of the Proposed Scheme

5.1.1. Reversibility. Figure 6(a) shows an original VQ-decoded image $\mathbf{I}$ for *Lena* sized 512×512, the length L of corresponding VQ codebook $\mathbf{C}$ is 256. In this experiment, the parameter σ in equation (6) was set as 1, and α was equal to 226 accordingly. The VQ-decoded, encrypted image with index permutation and codebook encryption is shown in Figure 6(b), which is the result through decoding $\mathbf{T}_e$ with $\mathbf{C}_e$. It can be observed that the contents of the original VQ-decoded image $\mathbf{I}$ are effectively masked after encryption. Figure 6(c) shows the VQ decoded, encrypted image after data embedding, which is the result through decoding $\mathbf{T}_{ew}$ with $\mathbf{C}_e$. The hiding capacity ζ was 12954 bits. Figure 6(d) is the directly decrypted image $\mathbf{I}_d$ for Figure 6(c), which is the result through decoding $\mathbf{T}_d$ with $\mathbf{C}$. PSNR of the directly decrypted result $\mathbf{I}_d$ in Figure 6(d) is 41.80 dB with respect to the original VQ-decoded image $\mathbf{I}$ in Figure 6(a). Recovered image, i.e., the result through decoding $\mathbf{T}$ with $\mathbf{C}$, is exactly

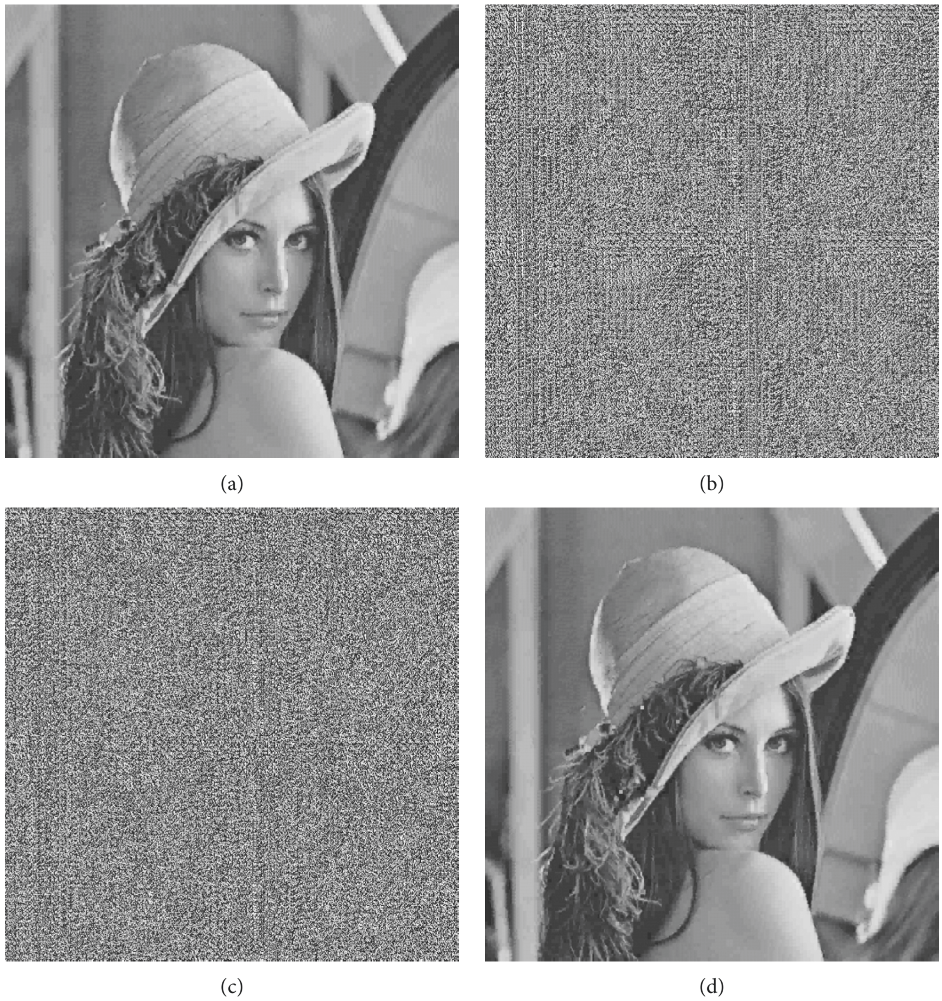


FIGURE 6: Results of the proposed scheme for image *Lena*. (a) Original VQ-image with decoding (T) by **C**. (b) Encrypted VQ-image with decoding $\mathbf{T}_e$ by $\mathbf{C}_e$. (c) Marked, encrypted VQ-image with decoding $\mathbf{T}_{ew}$ by $\mathbf{C}_e$ ($\zeta = 12954$ bits). (d) Directly decrypted VQ-image with decoding $\mathbf{T}_d$ by (C) (PSNR = 41.80 dB).

the same as **I**, i.e., PSNR = +∞, which demonstrates the reversibility of our scheme.

5.1.2. Hiding Capacity. Figure 7 shows four standard test images sized 512×512, including *Airplane*, *Baboon*, *Lena*, and *Peppers*. VQ compression was conducted for these four images, and the sizes L of the adopted codebooks can be 128, 256, 512, and 1024. Figure 8 shows hiding capacities ζ of our scheme for the four VQ-encoded images after encryption ($\sigma = 1$), in which (a)–(d) corresponds to the VQ codebook sizes $L = 128$, 256, 512, and 1024, respectively. Note that the abscissa denotes the different coefficient vectors $\boldsymbol{\mu}$ and the ordinate denotes corresponding hiding capacities ζ (bits). It can be observed that the codebook with larger size L can generally obtain greater hiding capacity ζ than the codebook with smaller size, since a large-size codebook can lead to more index groups for data embedding based on more VQ indices with lower occurrence numbers.

Besides the four images in Figure 7, Table 3 lists the largest hiding capacities with the optimal coefficient vectors $\boldsymbol{\mu}^{(i*)}$ of our scheme for more images. Actually, for different images, the hiding capacity of our scheme is mainly related with two aspects: (1) the value of $(L - \alpha + (1)$ under a given parameter σ, i.e., the number of the low-occurrence indices $\{c_\alpha, c_{\alpha+1}, \ldots, c_L\}$ and (2) the occurrence number vector $\mathbf{f} = [\gamma_1, \gamma_2, \ldots, \gamma_\beta]$, i.e., occurrence numbers of high-occurrence indices $\{c_1, c_2, \ldots, c_\beta\}$ in $\mathbf{T}_e$, see equations (14)–(16). Figure 9 shows the histograms of VQ indices that are sorted in the descending order according to occurrence numbers for the two images *Airplane* and *Baboon* ($L = 512$). We can clearly observe that the index histogram distribution of *Airplane* is more concentrated than that of Baboon, which means that the number of low-occurrence indices $\{c_\alpha, c_{\alpha+1}, \ldots, c_L\}$ and the occurrence numbers of high-occurrence indices $\{c_1, c_2, \ldots, c_\beta\}$ for *Airplane* are greater than those of *Baboon*. Correspondingly, it can be found from Table 3 that the hiding capacity for *Airplane* is significantly greater than that of *Baboon*. In addition, we also conducted experiments on the UCID image database [45], which consists of 1338 distinct images with the sizes of 512×384 and 384×512, see the last row of Table 3. For color images in the UCID

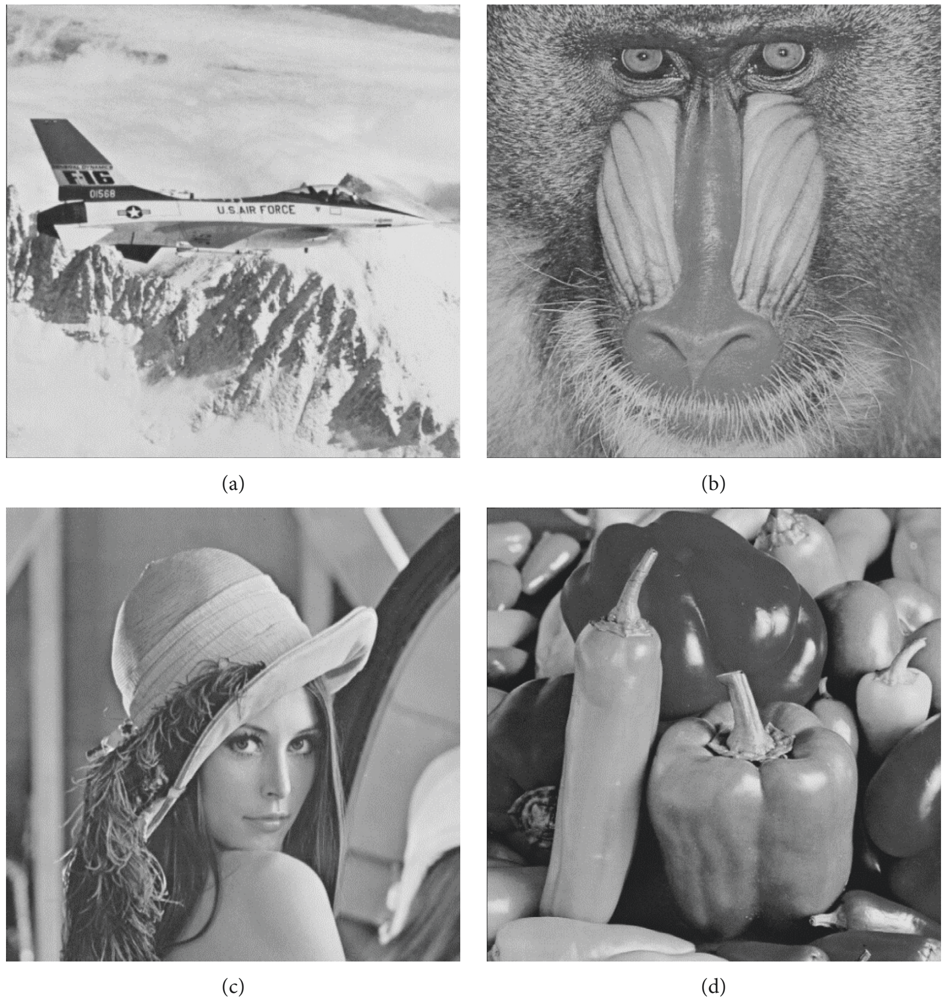


Figure 7: Four standard test images. (a) Airplane. (b) Baboon. (c) Lena. (d) Peppers.

database, the luminance components were applied for test, and the average hiding capacities are 6328 bits, 14649 bits, 21819 bits, and 29643 bits, when L is equal to 128, 256, 512, and 1024, respectively.

5.1.3. Visual Quality of Directly-Decrypted Image. As described previously, on the receiver side, the marked, encrypted index table $\mathbf{T}_{\mathrm{ew}}$, is scanned in raster-scanning order, and if the current scanning index $T_{x,y}'$ is equal to one of low-occurrence indices in an index group, i.e., $\{c_s^{(j,1)}, c_s^{(j,2)}, \ldots, c_s^{(j,2^v-1)}\}$, is equal to one of low-occurrence is modified as the corresponding the high-occurrence index $c_g^{(j)}$ in the same group, $j = 1, 2, \ldots, \beta$. After all indices in $\mathbf{T}_{\mathrm{ew}}$ are performed, $\mathbf{T}_{\mathrm{ew}}$ is changed as $\mathbf{T}_e'$. Then, after decrypting $\mathbf{T}_e'$, a directly-decrypted index table $\mathbf{T}_{\mathrm{d}}$ is generated, and the directly-decrypted image $\mathbf{I}_{\mathrm{d}}$ is obtained with decoding $\mathbf{T}_{\mathrm{d}}$ by VQ codebook $\mathbf{C}$.

Visual quality of directly-decrypted image $\mathbf{I}_{\mathrm{d}}$ for the proposed scheme is given in Table 4. As it reveals, PSNR values of directly-decrypted images decrease when the size L of VQ codebook increases. When the codebook size L increases, there may appear more VQ indices whose occurrence numbers are low ($\leq \sigma$) but non-zeros in the index table $\mathbf{T}_{\mathrm{e}}$. These indices should be replaced with the index c_{m} randomly selected from the set $\mathbf{\Omega}$ before data embedding, however, these indices cannot be recovered during image decryption, which causes the distortions in the directly-decrypted image $\mathbf{I}_{\mathrm{d}}$ with respect to the original VQ-decoded image $\mathbf{I}$. That is to say, larger codebook size L leads to more indices with non-zero, lower occurrence numbers that cannot be recovered after image decryption, thereby, lower PSNR of $\mathbf{I}_d$.

5.2. Performance Influence of Parameter σ. The parameter σ in equation (6) determines how many indices with lower occurrence numbers can be utilized in index grouping and data embedding; hence, the parameter σ affects the performance of hiding capacity and PSNR for the directly decrypted image, which are demonstrated in Figures 10 and 11, respectively. We can find that, with the increase of σ, the hiding capacity of our scheme increases, while the PSNR of directly decrypted image decreases. Because larger σ involves more indices with lower occurrence numbers for index grouping and data embedding, the

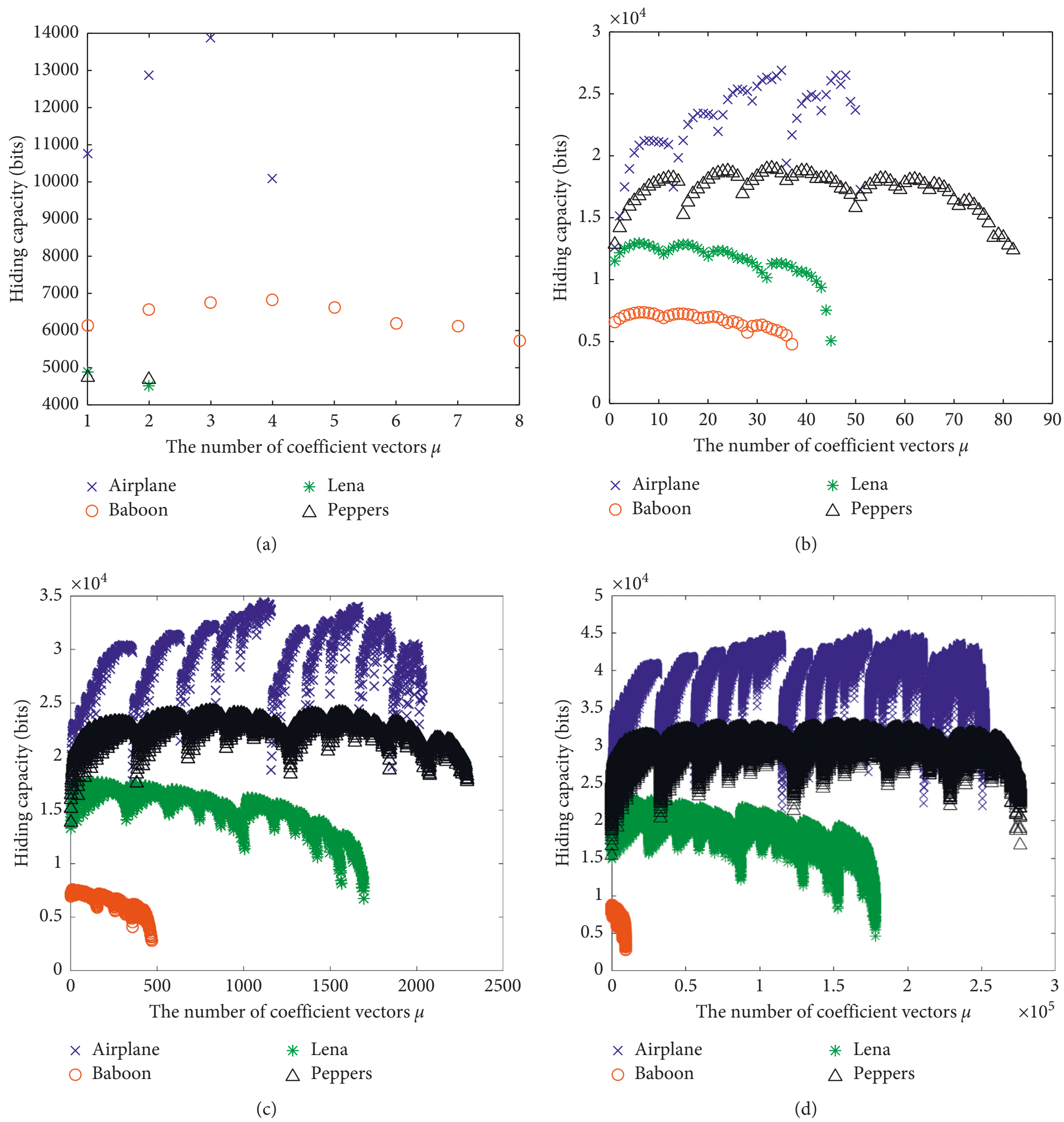


Figure 8: Hiding capacities with respect to different coefficient vectors $\boldsymbol{\mu}$ under four kinds of codebook sizes L ($\sigma = 1$). (a) $L = 128$. (b) $L = 256$. (c) $L = 512$. (d) $L = 1024$.

Table 3: The largest hiding capacities with the optimal coefficient vectors and $\sigma = 1$ (bits).

Images	$L = 128$	$L = 256$	$L = 512$	$L = 1024$
Aerial	4902	18004	24984	32209
Baboon	6835	7360	7571	8800
Barbara	3324	10439	12566	16765
Crowd	7614	21558	24359	31859
Peppers	4738	18990	24287	32834
Airplane	13855	26876	34424	45128
Lena	4889	12954	17624	23171
UCID	6328	14649	21819	29643

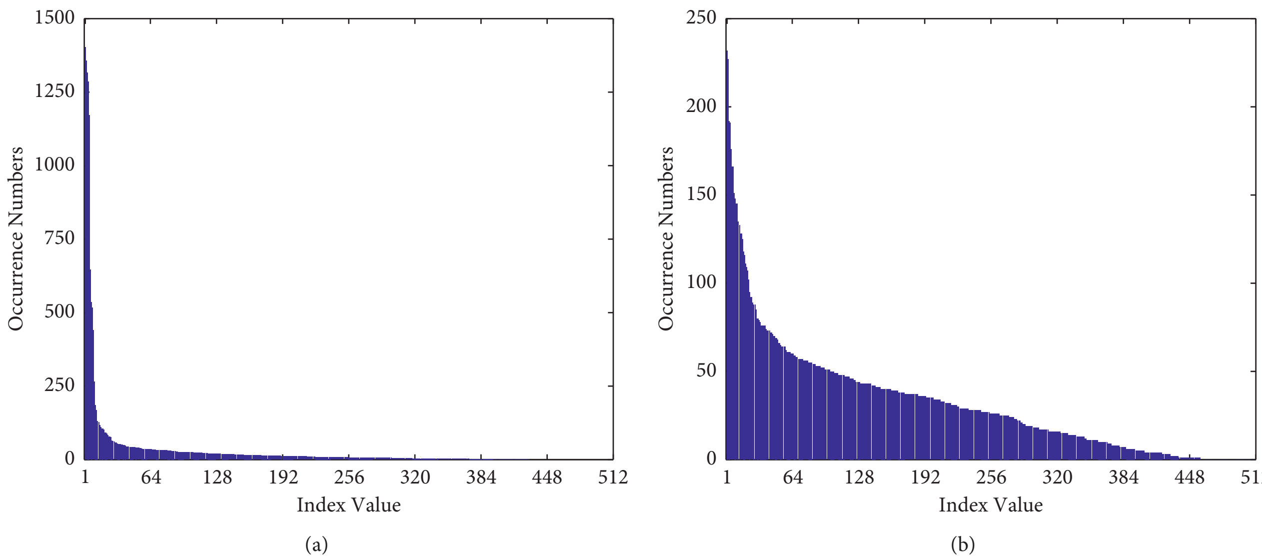


FIGURE 9: Histograms of sorted VQ indices ($L = 512$). (a) Airplane, (b) Baboon.

TABLE 4: PSNR of directly-decrypted image $\mathbf{I}_d$ (dB).

Images	$L = 128$	$L = 256$	$L = 512$	$L = 1024$
Aerial	45.20	38.10	32.00	28.10
Baboon	50.40	42.80	39.50	37.00
Barbara	45.90	45.00	38.50	35.40
Crowd	62.60	43.10	38.40	33.00
Peppers	53.60	42.70	38.20	33.20
Airplane	43.90	40.30	36.80	30.50
Lena	49.10	41.80	38.60	31.60
UCID	44.91	40.73	35.36	30.13

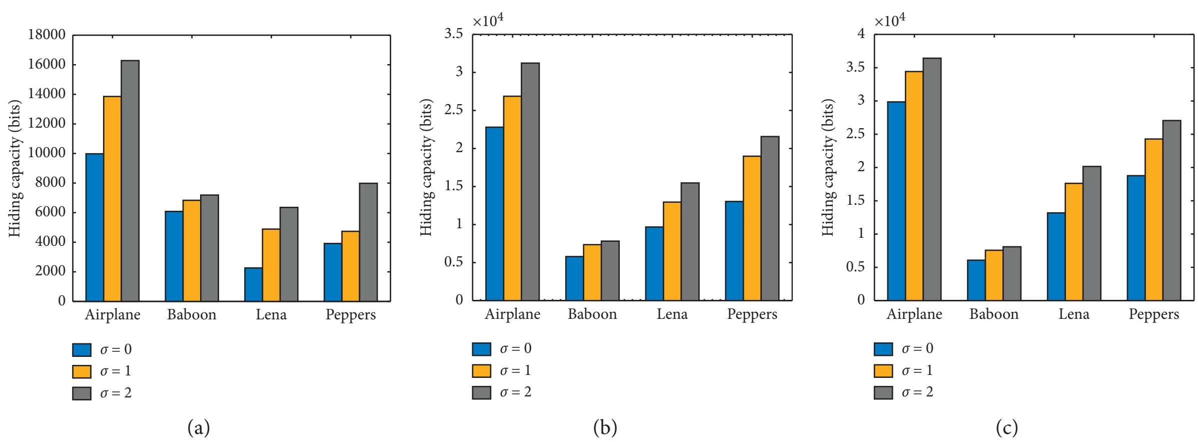
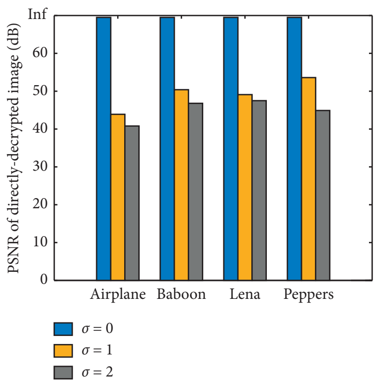
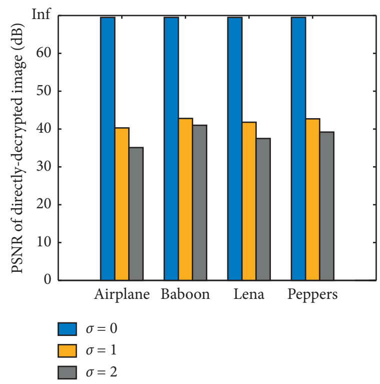


FIGURE 10: Hiding capacity under different values of parameter σ (bits). (a) $L = 128$. (b) $L = 256$. (c) $L = 512$.

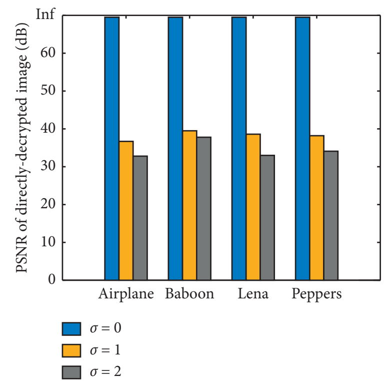
hiding capability becomes greater with the increase of parameter σ, see Figure 10. On the other hand, when σ is equal to 0, the occurrence numbers of the $L - \alpha + 1$ indices $\{c_\alpha, c_{\alpha+1}, \ldots, c_L\}$ are all zeros and no indices are required to be changed as c_m before data embedding; thus, after image decryption, original index table $\mathbf{T}$ can be obtained since $\mathbf{T}_e'$ is equal to $\mathbf{T}_e$, and the directly decrypted image $\mathbf{I}_d$ is exactly the same as $\mathbf{I}$, i.e., PSNR = inf. However, with the increase of σ, the value of α decreases, and more indices with nonzero occurrence numbers may be included in $\mathbf{\Phi}$, which are required to be changed as c_m before data embedding and cannot be recovered after image decryption, thereby, leading to lower PSNR of directly decrypted image $\mathbf{I}_d$, see Figure 11.



(a)



(b)



(c)

Figure 11: PSNR of directly decrypted image $(I)_d$ under different values of parameter σ (dB). (a) $L = 128$. (b) $L = 256$. (c) $L = 512$.

Table 5: Comparisons of embedding rate between the proposed scheme with Refs. [22, 26, 40–42].

Schemes	Airplane	Baboon	Lena	Peppers
Zhang's scheme [22]	0.0500	0.0500	0.0500	0.0500
Qian and Zhang's scheme [26]	0.2952	0.2952	0.2952	0.2952
Yin et al.'s scheme [40]	0.1647	0.8767	0.2919	0.3060
Qian et al.'s scheme [41]	0.0565	0.1151	0.0559	0.0649
Wu et al.'s scheme [42]	0.0555	0.5730	0.0204	0.3420
Proposed scheme ($L = 128$, $\sigma = 0$)	0.6091	0.3713	0.1381	0.2388
Proposed scheme ($L = 128$, $\sigma = 1$)	0.8456	0.4171	0.2984	0.2891
Proposed scheme ($L = 256$, $\sigma = 0$)	1.3919	0.3536	0.5909	0.7957
Proposed scheme ($L = 256$, $\sigma = 1$)	1.6403	0.4492	0.7906	1.1590
Proposed scheme ($L = 512$, $\sigma = 0$)	1.8227	0.3710	0.8048	1.1459
Proposed scheme ($L = 512$, $\sigma = 1$)	2.1010	0.4620	1.0756	1.4823
Proposed scheme ($L = 1024$, $\sigma = 0$)	2.4225	0.4358	1.1024	1.6953
Proposed scheme ($L = 1024$, $\sigma = 1$)	2.7544	0.5371	1.4142	2.0040

5.3. Comparison with State-of-the-Art Schemes. Since there are few RDHEI schemes for VQ-encoded images, we chose the other five typical RDHEI schemes, i.e., Zhang's scheme [22], Qian and Zhang's scheme [26], Yin et al.'s scheme [40], Qian et al.'s scheme [41], and Wu et al.'s scheme [42], for comparing the performance of embedding rate. In detail, the RDHEI schemes [22, 26] focused on uncompressed gray scale images, the schemes [40, 41] were designed for JPEG-encoded images, and the scheme [42] was applied in palette color images. As for the proposed scheme, we utilized four kinds of VQ codebooks with sizes of 128, 256, 512, and 1024, and the parameter σ was set to 0 and 1. It was worth noting that we used the unit of bpi (bits per index) to represent the embedding rate for our RDHEI scheme of VQ-encoded images and the unit of bpp (bits per pixel) for other schemes. Comparison results for the four standard images are given in Table 5. It can be observed from the results that our scheme generally has a competitive performance of embedding rate compared with the schemes in Refs. [22, 26, 40–42].

6. Conclusions

Reversible data hiding can be used in many scenarios like security and forensics. In this work, we focus on separable reversible data hiding in encrypted VQ-encoded images, which can achieve high hiding capacity and satisfactory image quality simultaneously. In order to protect the privacy of image contents, content-owner encrypts VQ codebook and index table with stream-cipher and permutation, respectively, and then sends the encrypted, VQ-encoded image to the data hider. In our baseline data-embedding method, the data-hider constructs index groups by grouping one high-occurrence index with one low-occurrence index; while in our optimized method, one high-occurrence index can be grouped with multiple low-occurrence indices to achieve greater hiding capacity. Through further optimizing the coefficient vector for different types of index groups, the optimal hiding capacity can be obtained by modifying the high-occurrence index into the corresponding indices in the same group according to the current to-be-embedded bits. Overall, more concentrated histogram of VQ indices leads to greater hiding capacity, and larger codebook leads to greater hiding capacity but lower directly decrypted image quality. Separable operations of data extraction, image decryption, and image recovery can be realized on the receiver side based on the availability of the encryption and data-hiding keys. Experimental results demonstrate the reversibility, security, hiding capacity, and parameter influence of our scheme, and the superiority compared with some state-of-the-art schemes. In the future work, we will further investigate the RDH for other types of encrypted data, such as video and audio.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61902239, 62172280, and 62172281, in part by the Natural Science Foundation of Shanghai under Grant 21ZR1444600, in part by the Natural Science Foundation of Shandong under Grant ZR2020MF054, in part by the STCSM Capability Construction Project for Shanghai Municipal Universities under Grant 20060502300, in part by the Shandong Provincial Natural Science Foundation (ZR2019BF017), Major Scientific and Technological Innovation Projects of Shandong Province (2019JZZY010127, 2019JZZY010132, and 2019JZZY010201), and in part by Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security under Grant MIMS20-03.

References

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

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

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

[4] Y.-Q. Shi, X. Li, X. Zhang, H.-T. Wu, and B. Ma, "Reversible data hiding: advances in the past two decades," *IEEE Access*, vol. 4, pp. 3210–3237, 2016.

[5] H. Yao, C. Qin, Z. Tang, and Y. Tian, "Guided filtering based color image reversible data hiding," *Journal of Visual Communication and Image Representation*, vol. 43, pp. 152–163, 2017.

[6] F. Huang, X. Qu, H. J. Kim, and J. Huang, "Reversible data hiding in JPEG Images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1610–1621, 2016.

[7] W. Lu, Y. J. Xue, Y. L. Yeung, H. M. Liu, J. W. Huang, and Y. Q. Shi, "Secure halftone image steganography based on pixel density transition," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1137–1149, 2021.

[8] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: a review," *IEEE Transactions on Communications*, vol. 36, no. 8, pp. 957–971, 1988.

[9] C.-C. Chang, Y.-P. Hsieh, and C.-Y. Lin, "Lossless data embedding with high embedding capacity based on declustering for VQ-compressed codes," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 341–349, 2007.

[10] J.-D. Lee, Y.-H. Chiou, and J.-M. Guo, "Reversible data hiding based on histogram modification of SMVQ indices," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 638–648, 2010.

[11] C. Qin, C.-C. Chang, and Y.-C. Chen, "Efficient reversible data hiding for VQ-compressed images based on index mapping mechanism," *Signal Processing*, vol. 93, no. 9, pp. 2687–2695, 2013.

[12] T. D. Kieu and S. Ramroach, "A reversible steganographic scheme for VQ indices based on joint neighboring coding," *Expert Systems with Applications*, vol. 42, no. 2, pp. 713–722, 2015.

[13] C.-C. Lin, X.-L. Liu, and S.-M. Yuan, "Reversible data hiding for VQ-compressed images based on search-order coding and state-codebook mapping," *Information Sciences*, vol. 293, no. 1, pp. 314–326, 2015.

[14] C. Qin and Y.-C. Hu, "Reversible data hiding in VQ index table with lossless coding and adaptive switching mechanism," *Signal Processing*, vol. 129, pp. 48–55, 2016.

[15] P. Rahmanl and G. Dastghaibyfard, "Two reversible data hiding schemes for VQ-compressed images based on index coding," *IET Image Processing*, vol. 12, no. 7, pp. 1195–1203, 2018.

[16] Z. Pan and L. Wang, "Novel reversible data hiding scheme for Two-stage VQ compressed images based on search-order coding," *Journal of Visual Communication and Image Representation*, vol. 50, pp. 186–198, 2018.

[17] W. Puech, M. Chaumont, and O. Strauss, "A reversible data hiding method for encrypted images," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 6819, pp. 1–9, 2008.

[18] C. Qin, Q. Zhou, F. Cao, J. Dong, and X. Zhang, "Flexible lossy compression for selective encrypted image with image inpainting," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 11, pp. 3341–3355, 2019.

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

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

[21] 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.

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

[23] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang, "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.

[24] F. Huang, J. Huang, and Y.-Q. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2777–2789, 2016.

[25] Y. Fu, P. Kong, H. Yao, Z. Tang, and C. Qin, "Effective reversible data hiding in encrypted image with adaptive encoding strategy," *Information Sciences*, vol. 494, pp. 21–36, https://www.sciencedirect.com/science/article/abs/pii/S0020025519303561?via%3Dihub, 2019.

[26] Z. Qian and X. Zhang, "Reversible data hiding in encrypted images with distributed source encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 636–646, 2016.

[27] C. Qin, W. Zhang, F. Cao, X. Zhang, and C.-C. Chang, "Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection," *Signal Processing*, vol. 153, pp. 109–122, 2018.

[28] S. Yi and Y. Zhou, "Separable and reversible data hiding in encrypted images using parametric binary tree labeling," *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 51–64, 2019.

[29] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 553–562, 2013.

[30] W. Zhang, K. Ma, and N. Yu, "Reversibility improved data hiding in encrypted images," *Signal Processing*, vol. 94, no. 1, pp. 118–127, 2014.

[31] X. Cao, L. Du, X. Wei, D. Meng, and X. Guo, "High capacity reversible data hiding in encrypted images by patch-level sparse representation," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1132–1143, 2016.

[32] P. Puteaux and W. Puech, "An efficient MSB prediction-based method for high-capacity reversible data hiding in encrypted images," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1670–1681, 2018.

[33] Z. Yin, Y. Xiang, and X. Zhang, "Reversible data hiding in encrypted images based on multi-MSB prediction and Huffman coding," *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 874–884, 2020.

[34] Y. Q. Wu, Y. Z. Xiang, Y. T. Guo, J. Tang, and Z. X. Yin, "An improved reversible data hiding in encrypted images using parametric binary tree labeling," *IEEE Transactions on Multimedia*, vol. 22, no. 8, pp. 1929–1938, 2020.

[35] Y.-C. Chen, C.-W. Shiu, and G. Horng, "Encrypted signal-based reversible data hiding with public key cryptosystem," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 1164–1170, 2014.

[36] X. Wu, B. Chen, and J. Weng, "Reversible data hiding for encrypted signals by homomorphic encryption and signal energy transfer," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 58–64, 2016.

[37] D. Xiao, Y. Xiang, H. Zheng, and Y. Wang, "Separable reversible data hiding in encrypted image based on pixel value ordering and additive homomorphism," *Journal of Visual Communication and Image Representation*, vol. 45, pp. 1–10, 2017.

[38] S. Xiang and X. Luo, "Reversible data hiding in homomorphic encrypted domain by mirroring ciphertext group," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 11, pp. 3099–3110, 2018.

[39] Z. Qian, H. Zhou, X. Zhang, and W. Zhang, "Separable reversible data hiding in encrypted JPEG bitstreams," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 6, pp. 1055–1067, 2018.

[40] Z. Yin, Y. Xiang, Z. Qian, and X. Zhang, "Unified data hiding and scrambling method for JPEG images," in *Proceedings of the 19th Pacific-Rim Conference on Multimedia*, pp. 373–383, Hefei, China, September 2018.

[41] Z. Qian, H. Xu, X. Luo, and X. Zhang, "New framework of reversible data hiding in encrypted JPEG bitstreams," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 351–362, 2019.

[42] H.-Z. Wu, Y.-Q. Shi, H.-X. Wang, and L.-N. Zhou, "Separable reversible data hiding for encrypted palette images with color partitioning and flipping verification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 8, pp. 1620–1631, 2017.

[43] F. Peng, Z.-X. Lin, X. Zhang, and M. Long, "Reversible data hiding in encrypted 2D vector graphics based on reversible mapping model for real numbers," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2400–2411, 2019.

[44] R. Jiang, H. Zhou, W. Zhang, and N. Yu, "Reversible data hiding in encrypted three-dimensional mesh models," *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 55–67, 2018.

[45] G. Schaefer and M. Stich, "Ucid – an uncompressed color image database," *Proceedings of SPIE in Storage and Retrieval Methods and Applications for Multimedia*, vol. 5307, pp. 472–480, 2004.