QDCT Encoding-Based Retrieval for Encrypted JPEG Images

Qiuju Ji[1], Peipeng Yu[1] and Zhihua Xia[1, *]

Abstract: A privacy-preserving search model for JPEG images is proposed in paper, which uses the bag-of-encrypted-words based on QDCT (Quaternion Discrete Cosine Transform) encoding. The JPEG image is obtained by a series of steps such as DCT (Discrete Cosine Transform) transformation, quantization, entropy coding, etc. In this paper, we firstly transform the images from spatial domain into quaternion domain. By analyzing the algebraic relationship between QDCT and DCT, a QDCT quantization table and QDTC coding for color images are proposed. Then the compressed image data is encrypted after the steps of block permutation, intra-block permutation, single table substitution and stream cipher. At last, the similarity between original image and query image can be measured by the Manhattan distance, which is calculated by two feature vectors with the model of bag-of -words on the cloud server side. The outcome shows good performance in security attack and retrieval accuracy.

Keywords: Image encryption, image retrieval, JPEG image, QDCT, bag-of-words.

1 Research background

With the rapid development of computer and Internet technologies, more and more devices are connected to the Internet to exchange data with other devices on the Internet. Image which is intuitive and vivid is a common form of data and carries more information than text mostly. Therefore, in the daily information exchange on the internet, image has become an indispensable part. Most devices such as computers, mobile phones, tablets, etc., are connected to the Internet and produce numbers of images every day. If the image is stored directly in the local device, it is easy to cause insufficient storage space or image loss, which is not expected.

The rise of cloud computing provides an opportunity for the storage of large amounts of data, which is currently the best choice for solving large numbers of image storage. Public clouds can solve the storage problem of images. However, it brings new security challenges. The image is stored on the public cloud, which means the image information is also directly exposed. In addition, the image is transmitted in the form of plaintext through an insecure common channel which means the transmission process also faces the danger of being intercepted by the attacker. Encryption is a common and valid method for the privacy and security of images on public clouds. After the image is generated by the local device, the user

[1] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China.

[*] Corresponding Author: Zhihua Xia. Email: xia_zhihua@163.com.

Received: 05 January 2020; Accepted: 13 May 2020.

JBD. doi:10.32604/jbd.2020.01004 www.techscience.com/journal/jbd

encrypts the image first, and then the image is uploaded as password text over a public channel to a public cloud server for storage.

In daily life, many image capture devices, such as smart phones, digital cameras, etc., produce images with a default format of JPEG. The JPEG image is obtained by a series of steps such as DCT (Discrete Cosine Transform) transformation, quantization, entropy coding, etc. The JPEG image obtained by compression has a file size much smaller than the original spatial image, and the storage is greatly saved. At the same time, the space does not affect the user's visual effects, so this image file format is widely used in real life. At present, most of the encryption algorithms for images use multi-channel independent encryption methods, which represents multiple color channels of the image as several grayscale images. It assists in encrypting each grayscale image separately by using the grayscale image encryption method, and then combines them. The final encrypted data will ignore the correlation between the color channels of the image, which means it has high computational complexity and wastes computing space, with the result of reducing the encryption efficiency to some extent. Therefore, the research of the combination of good effect of encryption and preserved image format in JPEG image retrieval has great practical significance.

In this paper, we proposed an image encryption algorithm for JPEG images based on QDCT encoding which supports feature extraction for image retrieval. The main contributions are summarized as follows:

1) An image encryption algorithm based on QDCT encoding is proposed for JPEG images. The images are firstly obtained by QDCT encoding instead of DCT encoding which means the original single channel processing feature has been changed into three channels which will be merged into one channel for processing. In this article, block shuffle, intra-block shuffle, and entropy code substitution are used to encrypt the images, which can protect the image data with small storage expansion. The encryption is mainly focused on the process of AC coefficients. JPEG images can better retain high-frequency AC coefficient information after QDCT encoding process. The quantization table of real and imaginary parts also enhances the degree of encryption which can achieve better encryption results and better retrieval accuracy. Furthermore, as the images are performed in quaternion domain, the original single-channel processing features become three channels and are merged into one channel for process, which means it fully takes the correlation between the three channels and the color information of the image fully consideration. In theory, it can increase the encryption efficiency and encryption effect. Then with the specially-designed encryption algorithm, the cloud server can directly extract features from the encrypted data for image retrieval, without additional communication with image owner.

2) We use the BOW model for the image retrieval in quaternion domain. We can extract the local features from the encrypted intermediate code in the JPEG compression process. We use visual text which generated by *K*-means clustering algorithm to construct the final feature vector, that is, the encrypted feature vector. Feature vectors' similarity can measure by Euclidean distance or Manhattan distance directly.

2 Related work

According to present image encryption, it mainly includes color image encryption algorithm based on index format, color image encryption algorithm based on color channel decomposition encryption and color image encryption algorithm based on quaternion representation.

Zhang et al. [Zhang and Karim (1999)] proposed a single-color image encryption algorithm based on double random phase pulling. In their algorithm, first transform the RGB color image into a representation in index format, and then use the double random phase encryption technology of the grayscale image to encrypt and transmit the integer matrix X. At the same time, use the color mapping matrix Map as the density to achieve Single-channel encrypted transmission of color images. In Joshi et al. [Joshi, Chandrashakher and Singh (2008)], this paper introduces a dual color image encryption algorithm based on dual random phase encryption. The integer matrix of two-color images is represented as a complex matrix using complex representation, and then phase modulation is performed in the space and transform domains respectively. At the same time, the color mapping matrix Map is used as Key. Although the above-mentioned strong encryption algorithm can realize single-channel encrypted transmission of color images, and is promoted from a single-color image to two- or four-color images, it is difficult to achieve more encrypted transmission of color images on this basis.

Joshi et al. [Joshi, Chandrashakher and Singh (2007)] proposed a single-channel encryption algorithm for color images. In their algorithm, the red, green, and blue H color channel components are separately encrypted. In Ho et al. [Ho and Myungjin (2014)] method, the image is encrypted once using double random phase and Fourier transform, and then 2292 is used for the complex data. The Hadamard transform is encoded again. At the same time, another representation form can be used for the decomposed color images, that is, stitching these single-channel images into a grayscale image. Although the above encryption algorithms based on color channel decomposition can realize the encrypted transmission of color images, there are some shortcomings. Firstly, there is a strong spectral relationship between the color channels of a color image. A color image is decomposed into a single channel separately to deal with the inherent relationship between them. Secondly, if an encryption method for grayscale images is used, then Some algorithms, such as double random phase encryption technology, will inevitably increase the number of phase masks and increase the storage space of the key in the encryption process. This makes it difficult to store and distribute the key. Thirdly, it is difficult to further popularize the above algorithms.

Fan et al. [Fan, Wang, Sun et al. (2015)] proposed a color image partial encryption algorithm based on QDCT (quaternion discrete cosine transform). By analyzing the algebraic relationship between QDCT and DCT, puts forward the quantization table of QCTT coefficients and QDCT coefficient coding for color images. Compared with the encryption algorithm in which the index format represents a color image, the quaternion-based encryption algorithm can avoid the color loss of the image during the transformation process. Compared with the single-channel encryption algorithm of color channel decomposition, the quaternion-based encryption algorithm is the integrity of the virtual to color image can also reduce the number of keys and facilitate the storage and distribution of keys.

The purpose of CBIR is to find images that are similar to queries. The similarity between images is generally judged by comparing the feature vectors extracted from the image content. How to extract the features is key point to get accurate results. After more than twenty years of development, CBIR techniques have shown its maturity in many real-world applications [Tang (1997)]. Many IT companies have provided the CBIR services, such as Google Images, Bing Images, Baidu Images, and so on. However, the existing technologies are mainly designed for plaintext images and cannot be directly applied to encrypted ones. It is worth noting that, in addition to the original image data, if it is not well protected, the image features will also leak information about the image content.

Searchable Encryption (SE) tries to protect the data by specially-designed encryption methods that meanwhile support data search over the cipher-text domain [Lian, Sun and Wang (2004)]. Recently, some privacy-preserving image retrieval schemes are proposed to support the searching on encrypted images. We divide these methods into two categories, feature-encryption based and image-encryption based schemes.

In the first case, the image owner extracts in advance the visual features from the images before image encryption. Then, the images can be protected by standard encryption technologies, and the features are protected by some specifically-designed methods. The encrypted features should still support the similarity comparison. Lu et al. [Lu, Swaminathan,Varna et al. (2009)] presented the first CBIR privacy protection scheme based on encrypted image database. The local features are extracted and clustered to generate visual words. Next, the inverted index is constructed, where each visual word is associated to a list of images and the occurrence frequency of this word in each of these images. Word frequency is protected by sequence-preserving encryption. The Jaccard distance is used to measure similarity between images. Lu et al. [Lu, Varna and Wu (2014)] studied bit plane randomization, random projection, random unitary coding and homomorphic encryption. Bit plane randomization and random unary coding support the calculation of hamming distance between encryption feature vectors. Xia et al. [Xia, Zhu, Sun et al. (2013)] utilized to secure *KNN* to protect the features and construct high level index by local sensitive hashing to improve the search efficiency. An encrypted image search model based on K nearest neighbor security algorithm is designed and a tree index is constructed by Yuan et al. [Yuan, Yu and Guo (2015)]. Zhang et al. [Zhang, Jung, Liu et al. (2017)] utilized the homomorphic encryption to protect the feature vectors. The distance between the encrypted feature vectors can be calculated in encryption domain. Xia et al. proposed a privacy CBIR scheme based on SIFT and bulldozer distance (EMD) is proposed by [Xia, Zhu, Sun et al. (2015)]. Huang et al. [Huang, Zhang, Pan et al. (2018)] proposed an interactive image retrieval scheme which keeps user’s search intention unrevealed. Cheng et al. [Cheng, Zhang , Yu et al. (2016)] proposed a CBIR scheme that supports image retrieval over multiple image owners. The image features are encrypted by a secure multi-party computation technique. The similarity between images is measured by a newly-designed distance criterion which may be quite different from the Euclidian distance. Qin et al. [Qin, Li, Xiang et al. (2019)] proposed an improved Harris algorithm is used to extract the image features. Then, Local Sensitive Hash algorithm is applied to construct the searchable index for the feature vectors. Wang et al. [Wang, Miao, Shen et al. (2019)] utilized the secure modular hashing to generate binary hash values to represent images. The similarity between the images can be measured by the distance of the vectors of binary values. Yan et al. [Yan, Chen and Jia (2019)] proposed to use Intel Software Guard Extension (SGX) to implement the secure similar image search. The images are encrypted by public key. The search operation is conducted in the so-called enclave where all the data are kept secure since no one, even the operation system can read or modify the data in it.

Shen et al. [Shen, Cheng, Zhu et al. (2018)] proposed a CBIR scheme that supports image retrieval over multiple image owners. The image feature is encrypted by a secure multi-party computation technique. Images' similarity is measured by a newly-designed distance criterion, which but may be quite different from the Euclidian distance. In addition, the computation complexity is high and the fully-trusted key management center is also not a desirable role in real applications.

In the second kind of schemes, the researchers tried to outsource the computation of the image feature extraction to further alleviate the image owner's burden. The key point of this kind of methods is to design a practicable image encryption algorithm which supports the feature extraction from the encrypted images for similarity measure.

Bellafqira et al. [Bellafqira, Coatrieux, Bouslimi et al.(2015); Two security CBIR schemes for homomorphic encryption are proposed by Bellafqira et al.: The SIFT [Bellafqira, Coatrieux, Bouslimi et al.(2016)] and discrete wavelet transform features [Bellafqira, Coatrieux, Bouslimi et al.(2015)] can be extracted from the encrypted image directly. Meanwhile, these features are also encrypted and can only support precise searches, because small changes in the plaintext will produce completely different password text. Xu et al. [Xu, Gong, Xiong et al. (2017)] presented a CBIR scheme for privacy protection based on partial encryption. Orthogonal decomposition of the image was divided into two parts. However, unencrypted parts of the data used for image retrieval can bring serious information leakage problem. Ferreira et al. [Ferreira, Rodrigues, Leitao et al. (2015); Ferreira, Rodrigues, Leitao et al. (2017)] designed Image Encryption Scheme that allow the direct extraction of image histogram. In this scheme, the color information is protected by pixel value permutation and the texture information is protected by row and column shuffle. It extracted HSV color histograms from encrypted images on the cloud server side. The similarity between images can be measured directly by the hamming distance. Cheng et al. [Cheng, Zhang, Yu et al. (2016)] proposed a JPEG image encryption scheme for secure image retrieval. The DC coefficients are encrypted by the stream cipher and permutation. The (r, v) pairs are also shuffled in blocks. The statistical features of (r, v) are calculated as block features and the similarity is measured by the block-wise feature comparison. Xu et al. [Xu, Gong, Xiong et al. (2017)] further fusion color histogram and AC coefficients histogram and got higher retrieval accuracy. Cheng et al. [Cheng, Zhang, Yu et al. (2016)] encrypted the DC coefficients, VLI code, and quantization tables to protect the JPEG images, leaving the length of the VLI code untouched. Thus, the intra-block, inter-block, and inter-component Markov features are extracted and used to train the SVM classifiers. Next, each image is mapped by the SVM classifiers into a short binary vector to facilitate the similarity calculation. Shen et al. [Shen, Cheng, Zhu et al. (2018)] constructed their scheme with the help of the tool in Secure multi-party computation (SMC), the different owners can secretly encrypt their feature vectors, and keep the sum of their feature vector same. The above methods are all ignoring that the owner can united as a group to provide the service. In fact, this kind of consideration is not only more practical, but also can relieve the computation cost in that the group are more like a normal owner when they are retrieval.

An outsourced CBIR scheme based on a new packet encryption word (BOEW) model is proposed by Xia et al. [Xia, Jiang, Liu et al. (2019)]. It uses the method of color value replacement, block replacement, and block pixel replacement to encrypt the images. The cloud server then calculates the local histogram based on the encrypted image block. Cluster all the local histograms together and use the cluster center as the encrypted visual word. In this way, we build an encrypted word package (BOEW) model and use a feature vector to represent each image. The similarity between images can be measured on the cloud server directly by the Manhattan distance between the feature vectors. However, the image cannot be restored to its original format.

3 System model and preliminaries

3.1 System model

The proposed scheme has three roles, namely image owner, user and cloud server as showed in Fig. 1. There are four primary steps performed by the owner and server respectively.

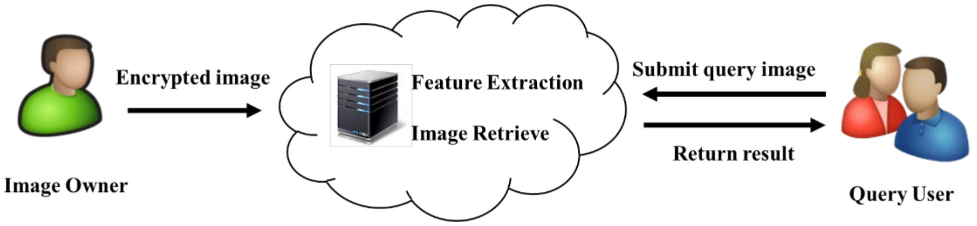


Figure 1: The system model of proposed scheme

Image owner firstly performs **ImgEnc,** uploads the encrypted images to cloud server. While **user** wants to query images in cloud database, the same image encryption methods must be conducted on the query image, and a corresponding **ImgDec** will be used to restore the retrieved images.

Cloud server uses the data uploaded by content owners to build an encrypted image database. **IndexGen** is then executed by cloud server to generate a secure index. In our scheme, cloud server also must extract features of all encrypted images in its database for subsequent retrieval, we will introduce our feature extraction method in 4.3. When the cloud server receives user's query request, **Similarity Measure** takes place between the query image and the entire database, and a similar image is returned to the user.

User wants to query images in cloud database, the same image encryption method will be performed as the **Image owner**. After the **Cloud Server** receives the request and does the operations between the query image and entire database, **user** gets the outcome.

3.2 Security model

A content owner plays the role of image query requester in a certain mode, as the only security risk is from an honest but curious cloud server which compliances with credit agreement but is easily access to user's data and any sensitive information. In our proposed scheme, authorized users i.e. content owners are perfectly believable and their operation is protocol compliant thus no information will be disclosed to cloud server. Although cloud server receives an encrypted query image and able to query similar images in the database, the image content is always encrypted and there is no information leakage. The security performance of proposed encryption algorithm will be thoroughly analyzed in Section 5, it is proved that cloud servers can barely crack encrypted data.

3.3 Overview of JPEG encoding based on QDCT

JPEG is a commonly used loss compression method for digital images. It is the most common format for image storing and transmitting due to its high compression ratio and little perceptible loss to human vision. Our format-compatible image encryption is designed to protect image data by encrypting the intermediate data in JPEG compression process. Therefore, we present a brief introduction about the JPEG compression to better explain the proposed encryption method.

We usually used the compression method for jpeg images before, a batch of steps is conducted in JPEG compression as illustrated in Fig. 2.

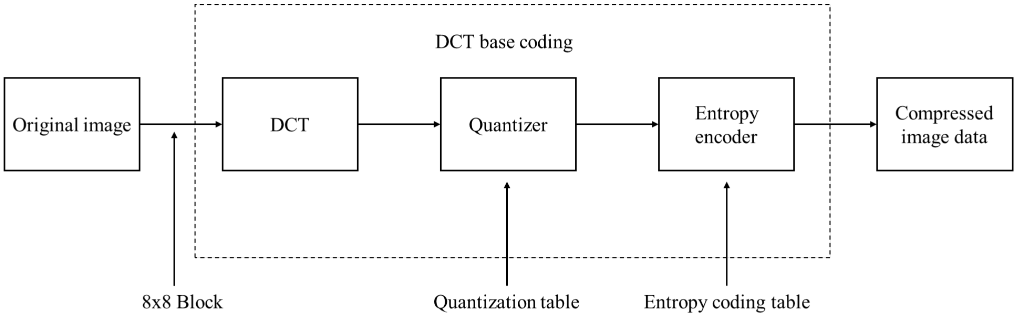


Figure 2: The framework of JPEG compression based on DCT

However, the compression did not fully consider the correlation between the three channels and the color information of the image as the original single-channel processing characteristics, meanwhile reducing the encryption efficiency and encryption effect. Then we consider do the compression based on QDCT. A batch of steps is conducted in JPEG compression as illustrated in Fig. 3. The entropy encoding is expanded in the field of quaternion based on jpeg DCT encoding. The entropy encoding process of the QDCT encoder is shown in Fig. 4.

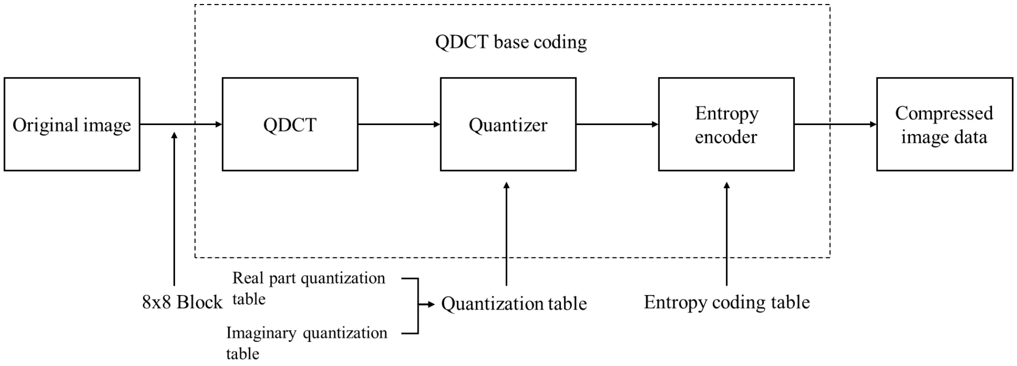


Figure 3: The framework of JPEG compression based on QDCT

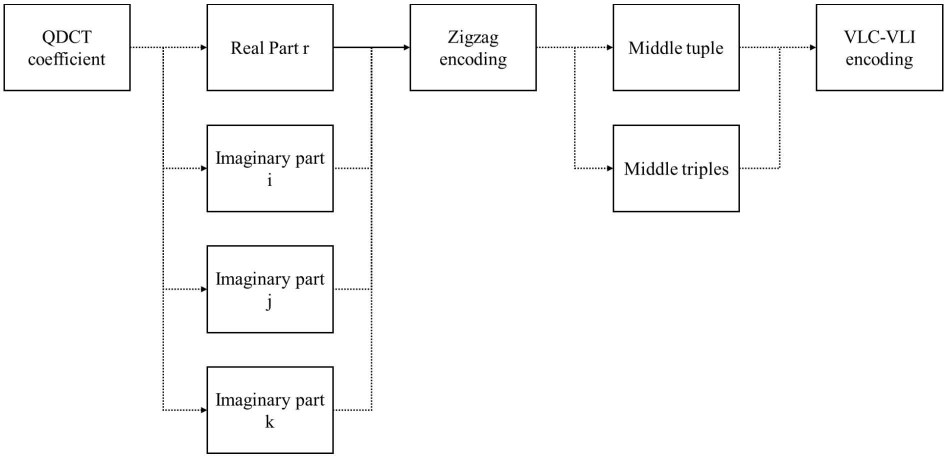


Figure 4: The entropy encoding process of the QDCT encoder

(1) **Color space transformation:** The color images are usually represented in RGB color space. The first step of JPEG encoding is to convert images from RGB into $\boldsymbol{YC_bC_r}$ color space, where **Y** stands for brightness, $\boldsymbol{C_b}$ and $\boldsymbol{C_r}$ stand for chromaticity and saturation. The human vision system is not so sensitive to the $\boldsymbol{C_r}$ and $\boldsymbol{C_b}$ components. Thus, the $\boldsymbol{YC_bC_r}$ color space allows greater compression without a significant effect on perceptual image quality by applying a higher compression on $\boldsymbol{C_r}$ and $\boldsymbol{C_b}$ components. Formally, the conversion can be formulated as follows,

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.2989 & 0.5866 & 0.1145 \\ -0.1687 & -0.3312 & 0.5000 \\ 0.5000 & -0.4183 & -0.0816 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

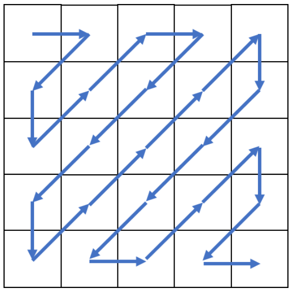


Figure 5: The process of Zig-Zag scan

(2) **Block splitting:** The three components in $\boldsymbol{YC_bC_r}$ color space are separately divided into the sub-blocks of 8×8 pixels before the subsequent QDCT. The sub-blocks are read from left and right, from top to bottom.

(3) **Quaternion Discrete Cosine Transform:** For $X \times Y$ size RGB image $f(x,y)$, x and y represent the row and column positions of the pixel respectively. $f(x,y)$ can be expressed as a quaternion as (4.1), in which $x \in [0, X-1]$, $y \in [0, Y-1]$.

$$f(x,y) = f_r(x,y) + f_i(x,y) \cdot i + f_j(x,y) \cdot j + f_k(x,y) \cdot k \tag{2}$$

In this formula, $f_r(x,y)$ equals zero and $f_i(x,y)$, $f_j(x,y)$, $f_k(x,y)$ represent the gray values of the three RGB color components at (x, y), respectively. Thus, the image $f(x,y)$ can be expressed as a quaternion matrix as (4.2).

$$f(x,y) = f_R(x,y) \cdot i + f_G(x,y) \cdot j + f_B(x,y) \cdot k \tag{3}$$

The three primary color components represent three imaginary parts and the real part is 0.

The QDCT positive transform is

$$FQDCT_q(p,s) = \alpha(p)\alpha(s\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}\mu_q \cdot f(x,y) \cdot N(p,s,x,y)) \tag{4}$$

(4) **Coefficient quantization:** Quantify the QDCT quaternion coefficient matrix, the real part coefficient C_r is quantified with T_{real} defined in Fan et al.'s method [Fan, Wang, Sun et al. (2015)], and the imaginary part coefficients C_i, C_j, C_k are quantized using T_{image} defined. The quantized coefficients are then encoded into binary bits.

(5) **Intermediate encoding:** The quantized coefficients are encoded by blocks. At this stage, the coefficients are encoded into the intermediate code. It is worth noting that our encryption method is also conducted on the intermediate code generated here. Zig-Zag scan coding is performed on the quantized QDCT real and imaginary coefficients to convert the two-dimensional matrix into a one-dimensional vector, the process like in Fig. 5. We use differential pulse code modulation encodes the DC coefficients of the real and imaginary coefficients, and expresses it in the form of an intermediate two-tuple (s,a). In (s,a), s is the correspond value in the VLI coding table as shown in Tab. 1 according to the amplitude between two DC coefficients and a is the amplitude between two DC coefficients. The Run-length coding is used to encode the AC coefficients of the real and imaginary coefficients, and is expressed in the form of an intermediate tuple (r,s,v). In (r,s,v), r denotes the number of zeros before a non-zero AC coefficient whose value equals v and s means the corresponding value in Tab. 1 according to v.

Table 1: Variable-Length Integer (VLI) coding table

Value v	Group index (number of bits)	Binary code
0	0	-
-1, 1	1	0,1
-3, -2, 2, 3	2	00, 01, 10, 11
-7, -6, -5, -4, 4, 5, 6, 7	3	000,001,010,011,100,101, 110,111
-15,…,-8, 8,…,15	4	0000, 0001,…,1110,1111
-31,…,-16, 16, 31	5	000000,…,111111
-63,…,-32, 32,…,63	6	0000000,…,1111111
-127,…, -64, 64,…, 127	7	00000000,…,11111111
-255,…, -128, 128,…,255	8	…
-511,…,-129,129,…,511	9	…
-1023,…,-512,512,…,1023	10	…

-2047,…,-1024,1024,…,2047	11	…
-4095,…,-2048,2048,…,4095	12	…
-8191,…,-4096,4096,…,8191	13	…
-16383,…,-8192,8192,…,16383	14	…
-32767,…,-16384,16384,…,32767	15	…

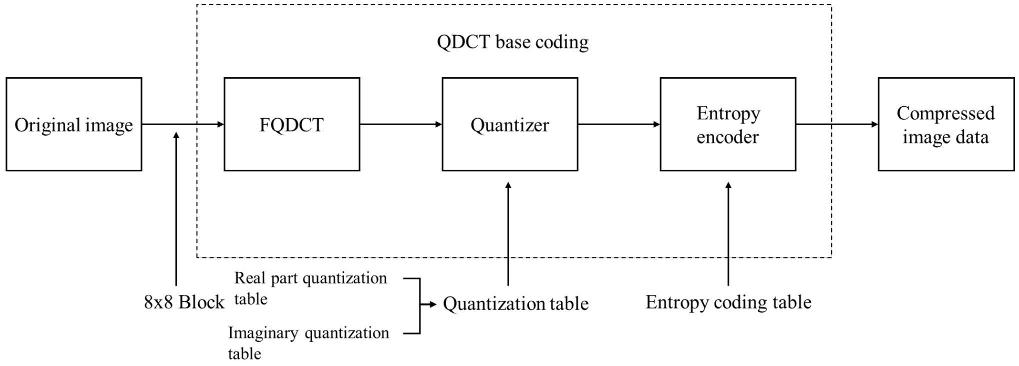


Figure 6: The process of entropy encoding based on QDCT

(6) **Entropy encoding:** The intermediate code generated above is further encoded by Variable-Length Integer (VLI) coding and Huffman coding. At first, the value v in (r, v) pair is denoted according to the VLI coding table as shown in Table 1. Then, the (r, v) pair will be converted to the triplet of run-length, group index, and binary code with VLI coding. We give flow chart in Fig. 6.

3.4 Bag-of-Words

In CBIR, features can be extracted in small local areas or in global images. Generally speaking, local features are more robust and retrieval results more accurate [Lu, Varna, Wu et al. (2014)]. A popular way to use local features is the word bag (BOW) pattern [Zhang and Cheng (2014)]. An image is analogous to a document. A word in an image can be defined as the eigenvector of the image block. Then the BOW model of the image is the histogram obtained from the feature vectors of all image blocks in the image. The model extracts local features from database images and then clusters them together. All the local features are represented by the closest visual word, and finally the image is represented by the histogram of the visual word. Specifically, the BOW model has three steps:

(1) **Feature extraction.** Assuming there are N images, the i-th image can be composed of $n\ (i)$ image patches, that is, it can be expressed by $n(i)$ feature vectors. In total, we can get $sum\ (n\ (i))$ feature vectors. Common features include Color histogram, SIFT, LBP, etc.

(2) **Dictionary Generation.** Using a clustering algorithm (such as the K-Means algorithm)

to construct a word list (dictionary) for the feature descriptors extracted in step (1). The feature descriptors are divided into K clusters so that the clusters have a high degree of similarity, and the similarity is low, the words with similar meanings are merged as the basic words in the word list. The number K of clustering categories is the number of basic words of the size of the entire visual dictionary.

(3) **Histogram Generation.** Many feature points are extracted from each image and can be replaced by words in the word list (visual words). By counting the number of times each word in the word list appears in the image, the image can be represented as a k-dimensional numerical vector.

4 The proposed method

4.1 Overview of the proposed project

The proposed project involves five algorithms which are executed by the image owner and cloud server in the initial and query images. The algorithms include image encryption, trap generation, and the image owner execute image decryption, the cloud server execute feature extraction and Search and the user execute image encryption, trap generation, and image decryption. The image owner encrypts the image and uploads it to the cloud server. The cloud server stores the encrypted image, extracts the image features from the encrypted image and builds an index. The user encrypts the image using the same encryption method as the image owner and then uploads it to the cloud server. The cloud server extracts feature from the encrypted image and compares them with the image features in the index table to send similar images to the querying user. The querying user decrypts the encrypted image to obtain the similar images.

In the initial stage, with an image database $\mathbb{I} = \{I_i\}_{i=1}^{n}$, the image owner runs image encryption to encrypt the image database, resulting in an encrypted image database $\mathcal{C} = \{C_i\}_{i=1}^{n}$. Next, the encrypted images are uploaded to the cloud server for storage. The cloud server runs index generation to build a secure index upon getting the encrypted image database.

4.2 Image encryption

In the proposed scheme, we use block permutation, intra-block permutation and a stream cipher to encrypt images.

Block permutation: As described in Subsection 3.2, the JPEG images are encoded by 8×8 blocks. As we use QDCT, we get 4 pairs of DC and AC coefficients. The (r, v) pairs are also organized by blocks, with a label of $(0,0)$ at the end of each block. In the proposed scheme, we shuffle such blocks to disrupt the texture information of image.

Inter-block Permutation: In each block, there are 4 parts of 63 AC coefficients, and it can be encoded to (r_1, v_1) $(r_2, v_2)(r_3, v_3)(r_4, v_4)$.Except the possible $(0,0)$ which appears at the end, the other (r_1, v_1) $(r_2, v_2)(r_3, v_3)(r_4, v_4)$ pairs within each block are shuffled to protect the image content.

Single Table substitution: To solve the feature leakage problem, we try to substitute the feature value. All the r and v in the other pairs are encrypted by the substitution. According to JPEG compression standard as described in Subsection 3.3, all the run-length r are within the range $[0,15]$, but the value v has an unlimited range, but in fact almost all v values will fall between $(-20, 20)$. Due to the requirements of JPEG format encoding, $(15,0)$ cannot be replaced with other numbers, so we replace the value of r between $(0,14)$ and the value of v between $(-20,-1) \cup (1,20)$, and v values exceeding the range are not converted. To facilitate decoding, we add $(0,0)$ as the terminator to the end of each block. For better understanding, we give an example in Fig. 7.

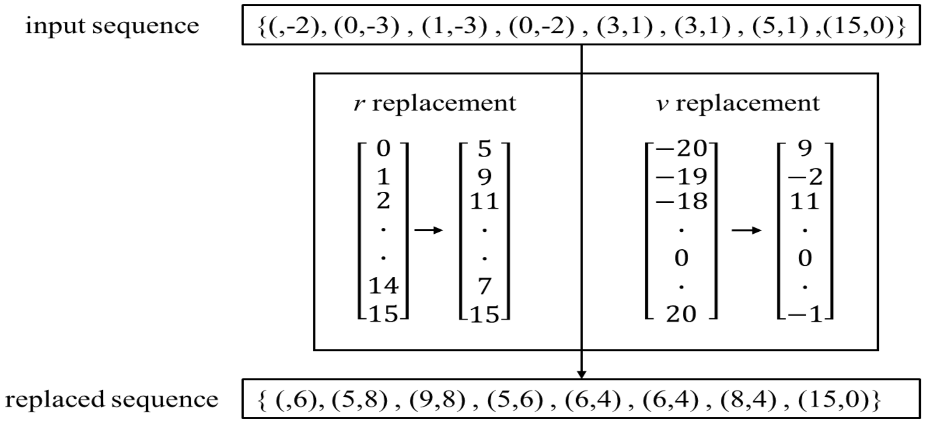

input sequence
{(,-2), (0,-3) , (1,-3) , (0,-2) , (3,1) , (3,1) , (5,1) ,(15,0)}
r replacement
v replacement
$\begin{bmatrix}0\\1\\2\\\cdot\\\cdot\\14\\15\end{bmatrix} \rightarrow \begin{bmatrix}5\\9\\11\\\cdot\\\cdot\\7\\15\end{bmatrix}$
$\begin{bmatrix}-20\\-19\\-18\\\cdot\\0\\\cdot\\20\end{bmatrix} \rightarrow \begin{bmatrix}9\\-2\\11\\\cdot\\0\\\cdot\\-1\end{bmatrix}$
replaced sequence
{ (,6), (5,8) , (9,8) , (5,6) , (6,4) , (6,4) , (8,4) , (15,0)}

Figure 7: The example of the process of single table substitution

Stream cipher: The DC coefficients are quite important in that it contains all the black and white information of color images. So, we use the stream cipher to encrypt the VLI code of the DC coefficients. With the QDCT process, we get two quantization table [Fan, Wang, Sun et al. (2015)]. For better safety, we encrypt 2 quantization tables by the stream cipher. At last, the data encrypted above are written into a bit-stream, which have the similar file format to the plaintext JPEG image.

4.3 Feature extraction

Assume that there are N images, each image can be composed of n image blocks, that means n feature vectors. The eigenvectors can be chosen according to the specific problem.

Based on the BOW model in Section 3, in our scheme, we let cloud server extract local features from the encrypted image data, and utilize the BOW model to calculate feature vector to represent images. It's the first try in quaternion domain which has good outcome in spatial domain [Xia, Jiang, Liu et al. (2019)]. Like the BOW model in spatial domain, there are also three steps in our method.

Feature extraction: This scheme calculates from each block a histogram as the local feature vector. At first, the encrypted JPEG images are parsed to get the 4 pairs of (r, v) pairs from each block. Since the cloud server holds no secret keys, (r_1, v_1) $(r_2, v_2)(r_3, v_3)(r_4, v_4)$ gets here are the encrypted ones by inter-block permutation. In this project, the local statistics for each block are measured by four vectors with five dimensions, defined as:

$$F_i = [n, m_i, k_i, q_i, z_i] \tag{5}$$

where n is the number of (r, v) pairs. M and k represent the mean and variance of

(r, v) within the block for the first value, respectively. q and z represent the mean and variance of (r, v) within the block for the second set of values, respectively. $i = 1,2,3,4$.

Dictionary Generation: A local histogram can be extracted from each 8×8 image block. Thus, the cloud server will get numbers of local histograms from a larger image database. Similar to the BOW in plaintext domain, the cloud server clusters these local features into k classes with the$k$-means clustering algorithm [Khan, Jeoti, Khan, et al(2010)]. The obtained Kcluster centers are defined as the encrypted visual words that make up the vocabulary.

Histogram Generation: In any image, each local histogram will be represented by its closest visual word in the vocabulary. The corresponding image is represented by calculating the appearance histogram of visual words, which is convenient for similarity calculation. Thus, each image is represented by four histograms $F_i = [n, m_i, k_i, q_i, z_i]$. In this way, images' similarity can be measured by the distance between the occurrence histograms of visual words.

4.4 Image search

In order to search similar images, the user encrypts the query image and submit it as the trapdoor to the cloud server. The cloud server extracts the occurrence histograms of visual words from the query image F_q and calculate the distance between F_q and each of F_i extracted from image database to measure the similarity of them. Then the cloud server returns the θ most similar images to the user.

5 Security analysis

The encryption mechanism proposed in this paper can even resist violent attacks, known plaintext attacks and statistical analysis of color histogram. For any image encrypted by this scheme, it is almost impossible for an attacker to infer the original image by exhaustive search. Therefore, this encryption method can prevent brute force attacks.

Since the proposed encryption method combines stream encryption and displacement encryption, exhaustive search is still the main method to guess the attackers to use several pairs of plain text images and encryption keys. As described in brute-force attack analysis, attackers using exhaustive search display only raw quantization tables encrypted by standard stream passwords, which can be time-consuming and laborious. In addition, in our encryption method, due to the use of QDCT coding method, the quantization table, DC and AC coefficients become 4, which further expands the space for exhaustive search and greatly increases the computational cost of the attacker. Based on the above reasons, the proposed encryption method is secure against known plaintext attacks.

Displacement operation can maintain local statistical invariance of AC coefficient before and after encryption. Using the immutability captured by the proposed 5-dimensional feature descriptor, the BOW model makes it easy to calculate the similarity between encrypted JPEG images without using any encryption keys on the cloud server. In this case, our retrieval scheme can support different encryption keys for different images, and the encryption keys are convertible. Therefore, from the perspective of the overall retrieval scheme, it is secure against selected plaintext attacks. Different images (or even the same image) may require different decryption keys throughout the retrieval scheme.

6 Experimental result

This section evaluates the performance of the proposed scheme from the aspects of encryption effectiveness, retrieval accuracy and efficiency. Corel Image Database1 contains 1000 images in JPEG format, each 384 29256 or 256 29384. In the experiment, we use the encryption mechanism proposed in this paper to encrypt all database images and select each encrypted image from the encrypted image database as the encrypted query image. The precision-recall curve and mAP are used to measure the retrieval accuracy performance of the proposed scheme. The further test of validity of different feature in proposed model based on QDCT, we also structure vocabulary with the 5-dimensional used in Cheng et al. and Khan et al.'s methods [Cheng, Zhang, Yu et al. (2016); Khan, Jeoti, Khan et al. (2010)].

6.1 Effectiveness of image encryptions

In our scheme, the images are encrypted by block permutation, intra-block permutation, single table substitution and stream cipher. We should notice that all the encryption methods only have a linear time complexity. As after the method of single table replacement, the jpeg images have been turned to jpeg bit-stream which can't change to the image. We performed four steps of encryption in the Corel database which has 1000 images.

6.2 Expansion of file size

In the proposed image encryption, the encoding method of QDCT and the replacement of the values of r and v will cause the JPEG image file size to expand. Specifically, due to the encoding method, there will be 4 DC and AC coefficients, including quantization tables in each part. In an unencrypted JPEG image, the run length r with a high probability of occurrence will be encoded by a short code according to the Huffman encoding algorithm. However, after replacement, there is no guarantee that high-frequency runs are encoded by short codes. In addition, most values v have smaller absolute values and will be encoded using short codes, which will also change after replacement. The above two reasons lead to the expansion of the file size. In Tab. 2, we compare the file extensions of our scheme with those of previous schemes that encrypt images in the spatial domains [Cheng, Zhang , Yu et al. (2016)] and [Lu, Varna and Wu (2014)]. This shows that the scheme is much less scalable. Since encrypted images are already stored in the cloud, users don't have much burden.

Table 2: The size of the encrypted image database

	Unencrypted	Encrypted by Our method	Encrypted by Cheng, et al.'s method (2016)	Encrypted by Lu et al.'s method (2014)
Size of Image Database	35.6 MB	152 MB	281 MB	287.5 MB

6.3 Retrieval accuracy

In our experiment, precision-recall curve and mean average precision (mAP) are used to measure the retrieval accuracy. We take the average precision-recall curve for all query images as a final performance measure of an image retrieval algorithm. The precision and recall are defined as follow:

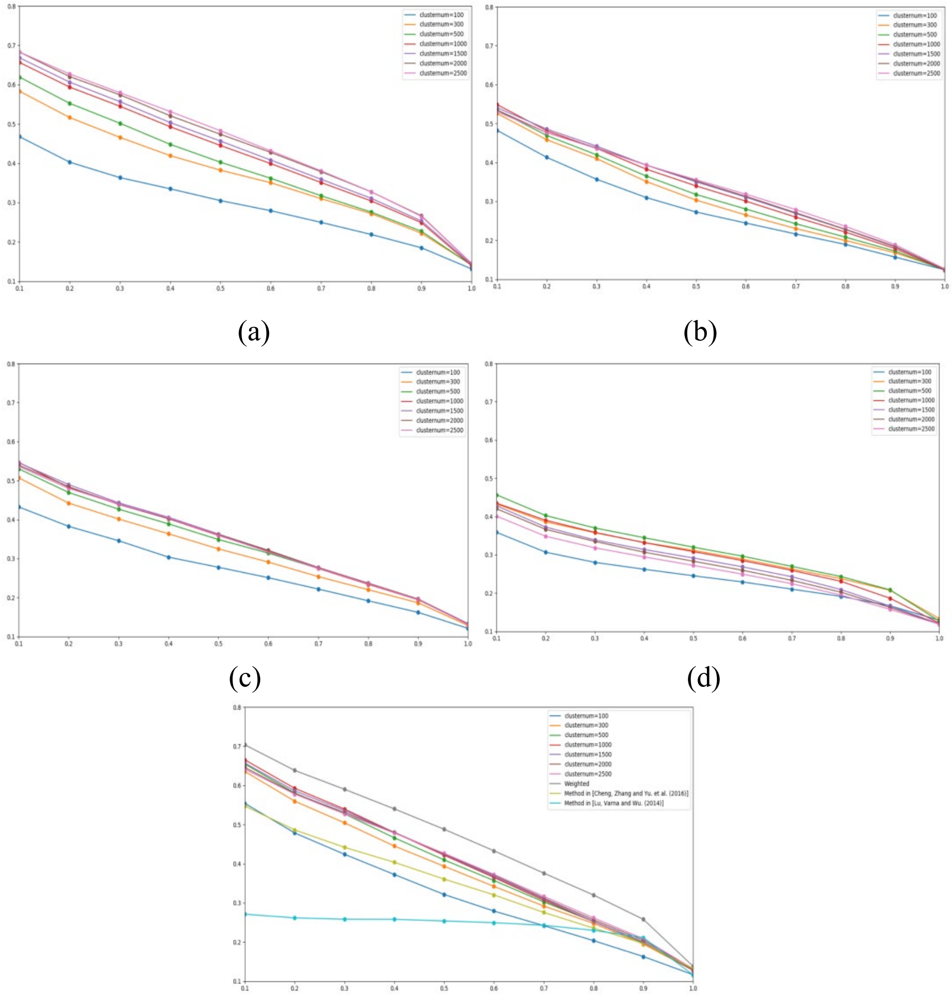
$$\text{Precision} = \frac{N_p}{N_r}, \quad Recall = \frac{N_p}{N_A} \tag{6}$$

where N_p is the number of returned positive images, N_r is the number of all returned images, and N_A is the number of all positive images in the database.

In our experiment, mAP values are defined as:

$$\text{mAP} = \frac{\sum_{n=1}^{N} \int_0^1 P_n(R_n) dR_n}{N} \tag{7}$$

where i means the number of precision and recall pairs.



(a)

(b)

(c)

(d)

(e)

Figure 8: Precision-recall curves of different cluster center and result of weighted method, [Cheng, Zhang, Yu et al. (2016)], [Lu, Varna and Wu (2014)]

Table 3: mAP value of different cluster numbers and weighted method [Cheng, Zhang, Yu et al. (2016); Lu, Varna and Wu (2014)]

CC Number mAP Component	100	300	500	1000	1500	2000	2500
r1	0.264	0.33	0.347	0.378	0.386	0.4	0.404
r2	0.246	0.271	0.28	0.294	0.299	0.299	0.301
r3	0.241	0.28	0.298	0.304	0.307	0.305	0.304
r4	0.213	0.267	0.275	0.263	0.248	0.242	0.232
ALL	0.282	0.336	0.349	0.355	0.355	0.355	0.354
Weighted				0.406			
Cheng, Zhang, Yu et al. (2016)				0.36			
Lu, Varna and Wu (2014)				0.22			

Fig. 8 gives the performance of the proposed method under different cluster centers ,weighted method and the previous method [Cheng, Zhang, Yu et al. (2016); Lu, Varna and Wu (2014)]. It can be seen that when the cluster centers up more than 300 and 1500 , our method result is better than Cheng et al. and Lu et al. 's methods [Cheng, Zhang, Yu et al. (2016); Lu, Varna and Wu (2014)]. When the cluster up to 1500, our method gets the best pr value. As each graph has 2304 rows, we don't need to consider them all when clustering. The four components consist different information, so we only need to extract a certain percentage of data to get similar clustering results. In our experiment, the data of each block is extracted in different weights. When we set the weights as (1, 0.2, 0.1, 0.3), we get the best result, which is show in Fig. 8. The clustering is iterated 500 times totally. Compared with the method in Cheng et al. [Cheng, Zhang, Yu et al. (2016)] and Lu et al. [Lu, Varna and Wu (2014)], our method uses much less time and gets better performance. The detail of time and mAP value with different component will give in Fig. 4 and Tab. 3.

Table 4: Time consumption of encryption

	Block permutation	Inter-block Permutation	Stream cipher	Value substitution	Total
Encryption(s)	0.03	0.38	0.13	1.16	1.7

Table 5: Time consumption of retrieval

Cluster center number	100	300	500	1000	1500	2000	2500
Retrieval(s)	6.14	6.15	616	6.21	6.24	6.25	6.26

6.4 Time consumption

In Tab. 4, we give the time consumption of encryption, and the time of retrieval is given in Tab.5. It could be seen that the time spent on each step of the experiment is not too much. The step of clustering can be performed automatically by the server in the background which means it does not affect the user which we didn't give the table here.

7 Conclusion

We perform image processing in the JPEG domain to ensure the compatibility while ensuring compression ratio. What's more, we extended the DCT transform to QDCT, and the original single channel processing feature became three channels and merged into one channel for processing, taking into account the correlation between the three channels and the color information of the image, which can theoretically increase the encryption efficiency and Encryption effect. In this article, our encryption is mainly focused on the processing of AC coefficients. JPEG images can better retain high-frequency AC coefficient information after QDCT processing. The quantization table of real and imaginary parts also strengthens the degree of encryption to achieve the encryption effect and retrieval accuracy are better, and the BOW model is combined, which will also save computing overhead. In future research, on one hand, we can consider other encryption strategies and feature extraction strategies, such as encrypt DC coefficients with some functional encryptions in that the DC coefficients have great information on retrieval. On the other hand, we may consider the system model that multiple Image owners trying to execute secure image retrieval together.

Funding Statement: This work is supported in part by the Jiangsu Basic Research Programs-Natural Science Foundation under grant numbers BK20181407, in part by the National Natural Science Foundation of China under grant numbers U1936118, 61672294, in part by Six peak talent project of Jiangsu Province (R2016L13), Qinglan Project of Jiangsu Province, and "333" project of Jiangsu Province, in part by the National Natural Science Foundation of China under grant numbers U1836208, 61702276, 61772283, 61602253, and 61601236, in part by National Key R \& D Program of China under grant 2018YFB1003205, in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, in part by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China. Zhihua Xia is supported by BK21+ program from the Ministry of Education of Korea.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Bellafqira, R.; Coatrieux, G.; Bouslimi, D.; Quellec, G. (2015): Content-based image retrieval in homomorphic encryption domain. *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2944-2947.

Bellafqira, R.; Coatrieux, G.; Bouslimi, D.; Quellec, G. (2016): An end to end secure

CBIR over encrypted medical database. *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2537-2540.

Cheng, H.; Zhang, X.; Yu, J. (2016): Encrypted JPEG image retrieval using block-wise feature comparison. *Journal of Visual Communication and Image Representation*, vol. 40, Part A, pp. 111-117.

Cheng, H.; Zhang, X.; Yu, J. (2016): Markov process based retrieval for encrypted JPEG images. *EURASIP Journal on Information Security*, no. 1.

Fan, J.; Wang, J.; Sun, X.; Li, T. (2015): Partial encryption of color image using quaternion discrete cosine transform. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, no. 10, pp. 171-190.

Ferreira, B.; Rodrigues, J.; Leitao, J. (2017): Practical privacy-preserving content-based retrieval in cloud image repositories. *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 784-798.

Ferreira, B.; Rodrigues, J.; Leitao, J.; Domingos, H. (2015): Privacy-preserving content-based image retrieval in the cloud. *IEEE 34th Symposium on Reliable Distributed Systems*, pp. 11-20.

Ho, L.; Myungjin, C. (2014): Double random phase encryption based orthogonal encoding technique for color images. *Journal of the Optical Society of Korea*, vol. 18, no. 2, pp. 129-133.

Huang, Y.; Zhang, J.; Pan, L. (2018): Privacy protection in interactive content based image retrieval. *IEEE Transactions on Dependable and Secure Computing*.

Joshi, M.; Chandrashakher; Singh, K. (2007): Color image encryption and decryption using fractional fourier transform. *Optics Communications*, vol. 279, no. 1, pp. 35-42.

Joshi, M.; Chandrashakher; Singh, K. (2008): Color image encryption and decryption for twin images in fractional fourier domain. *Optics Communications*, vol. 281, no. 23, pp. 5713-5720.

Khan, M. I.; Jeoti, V.; Khan, M. A. (2010): Perceptual encryption of JPEG compressed images using DCT coefficients and splitting of DC coefficients into bitplanes. *International Conference on Intelligent and Advanced Systems.*

Lian, S.; Sun, J.; Wang, Z. (2004): A novel image encryption scheme based-on JPEG encoding. *IEEE International Conference on Information Visualisation*.

Lu, W.; Swaminathan, A.; Varna, A. L.; Wu, M. (2009): Enabling search over encrypted multimedia databases. *IS&T/SPIE Electronic Imaging*, pp. 725418.

Lu, W.; Varna, A. L.; Wu, M. (2014): Confidentiality-preserving image search: a comparative study between homomorphic encryption and distance-preserving randomization. *Quality Control, Transactions*, vol. 2, no. 2, pp. 125-141.

Qin, J.; Li, H.; Xiang, X.; Tan, Y. (2019): An encrypted image retrieval method based on harris corner optimization and LSH in cloud computing. *IEEE Access*, vol. 7, pp. 24626-24633.

Shen, M.; Cheng, G.; Zhu, L. (2018): Content-based multi-source encrypted image retrieval in clouds with privacy preservation. *Future Generation Computer Systems*.

S0167739X17321969.

Tang, L. (1997): Methods for encrypting and decrypting MPEG video data efficiently. *ACM.*

Wang, Y.; Miao, M.; Shen, J. (2019): Towards efficient privacy-preserving encrypted image search in cloud computing. *Soft Computing*, vol. 23, pp. 2101-2112.

Xia, Z.; Jiang, L.; Liu, D. (2019): BOEW: a content-based image retrieval scheme using bag-of-words in cloud computing. *IEEE Transactions on Services Computing.*

Xia, Z.; Zhu, Y.; Sun, X.; Qin, Z. (2015): Towards privacy-preserving content-based image retrieval in cloud computing. *IEEE Transactions on Cloud Computing*, pp. 1.

Xia, Z.; Zhu, Y.; Sun, X.; Wang, J. (2013): A similarity search scheme over encrypted cloud images based on secure transformation. *International Journal of Future Generation Communication and Networking*, vol. 6, no. 6, pp. 71-80.

Xu, Y.; Gong, J.; Xiong, L. (2017): A privacy-preserving content-based image retrieval method in cloud environment. *Journal of Visual Communication and Image Representation*, vol. 43, pp. 164-172.

Yan, H.; Chen, Z.; Jia, C. (2019): SSIR: Secure similarity image retrieval in IoT. *Information Sciences*, vol. 479, pp. 153-163.

Yuan, J.; Yu, S.; Guo, L. (2015): SEISA: Secure and efficient encrypted image search with access control. *IEEE Conference on Computer Communications.*

Zhang, L.; Jung, T.; Liu, K.; Li, X. Y. (2017): Pic: Enable largescale privacy preserving content based image search on cloud. *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 11, pp. 3258-3271.

Zhang, S.; Karim, M. A. (1999): Color image encryption using double random phase encoding. *Microwave and Optical Technology Letters*, vol. 21, no. 5, pp. 318-323.

Zhang, X.; Cheng, H. (2014): Histogram-based retrieval for encrypted JPEG images. *IEEE China Summit & International Conference on Signal & Information Processing*.