A Privacy-Preserving JPEG Image Retrieval Scheme Using the Local Markov Feature and Bag-of-Words Model in Cloud Computing

Peipeng Yu, Jian Tang, Zhihua Xia, *Member, IEEE*, Zhetao Li, *Member, IEEE*, and Jian Weng

Abstract—The development of cloud computing attracts a great deal of image owners to upload their images to the cloud server to save the local storage. But privacy becomes a great concern to the owner. A forthright way is to encrypt the images before uploading, which, however, would obstruct the efficient usage of image, such as the Content-Based Image Retrieval (CBIR). In this paper, we propose a privacy-preserving JPEG image retrieval scheme. The image content is protected by a specially-designed image encryption method, which is compatible to JPEG compression and makes no expansion to the final JPEG files. Then, the encrypted JPEG files are uploaded to the cloud, and the cloud can directly extract the features from the encrypted JPEG files for searching similar images. Specifically, big-blocks are first assembled with adjacent 8×8 discrete cosine transform (DCT) coefficient blocks. Then, the big-blocks are permuted and the binary code of DCT coefficients are substituted, so as to disturb the content of image. After receiving the encrypted images, local Markov features are extracted from the encrypted big-blocks, and then the Bag-Of-Words (BOW) model is applied to construct a feature vector with these local features to represent the image, so as to provide the CBIR service to image owner. Experimental results and security analysis demonstrate the retrieval performance and security of our scheme.

Index Terms—Searchable encryption, privacy-preserving information retrieval, format-compatible encryption, bag-of-words

1 Introduction

Nowadays, a great number of images are generated every second. Generally, images are large in size and thus can use up the users' local storage quickly, especially for smartphones. To allow users to manage the images for more convenient storage and access, many Content-based image retrieval (CBIR) schemes are developed. As shown in Fig. 1, CBIR allows users to input a picture and find other pictures with the same or similar content, which has been a common scene in real life. Baidu, Google, and other search engines have launched the corresponding CBIR products, which provide a great convenience for image retrieval in large-scale data sets.

However, the first concern of such wonderful services is about the privacy [1]. For the image owners, the outsourcing of images implies the loss of control over their images which may contain sensitive information. Many incidents have revealed that the strong cloud computing services can also be hacked. A recent TechCrunch report showed that Instagram leaked 49 million users' sensitive data, exposing them to Internet [2]. The iCloud celebrity photo leakage also illustrated the danger of image outsourcing [3]. So, the cloud server can be never fully trusted. In addition, malicious internal employees can also be a great threat to the outsourced data. A general solution to this problem is to encrypt the images before outsourcing. But image encryption will impede the efficient utilization of image, including the Content-Based Image Retrieval (CBIR) service. Without a special design, the image owner needs to download, decrypt, and search the whole database when he needs to retrieve similar images to a query, which raises unacceptable communication, storage, and computation burdens to the owner.

Many Privacy-Preserving Content-Based Image Retrieval (PPCBIR) schemes have been designed to search similar images with privacy protected. The existing PPCBIR schemes

- *Peipeng Yu, Zhihua Xia, Zhetao Li, and Jian Weng are with the National & Local Joint Engineering Research Center of Network Security Detection and Protection Technology, Guangdong Provincial Key Laboratory of Data Security and Privacy Protection, Engineering Research Center of Trustworthy AI, Ministry of Education, College of Cyber Security, Jinan University, Guangzhou 510632, China. E-mail: {ypp865, xia_zhihua}@163.com, {liztchina, cryptjweng}@gmail.com.*
- *Jian Tang is with the School of Computer, Nanjing University of Information Science and Technology, Nanjing 210044, China. E-mail: j.tang@nuist.edu.cn.*

Manuscript received 24 October 2021; revised 21 October 2022; accepted 27 December 2022. Date of publication 2 January 2023; date of current version 6 September 2023.

This work was supported in part by the National Key R&D Program of China under Grants 2022YFB3103100, 2021YFB3101201, and 2020YFB1005600, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2019B1515120010, in part by the National Natural Science Foundation of China under Grants 62122032, 62032020, 62172233, 62102189, U1936118, 61932011, 61931004, 61825203, U1736203, and 61732021, in part by the Major Program of Guangdong Basic and Applied Research Project under Grant 2019B030302008, in part by the Six Peak Talent project of Jiangsu Province under Grant R2016L13, in part by the te Qinglan Project of Jiangsu Province , in part by the "333" project of Jiangsu Province , in part by the National Joint Engineering Research Center for Network Security Detection and Protection Technology , 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. The work of Zhihua Xia was supported by BK21+ program from the Ministry of Education of Korea. Peipeng Yu and Jian Tang contribute equally to this work.

(Corresponding author: Zhihua Xia.)

Recommended for acceptance by M. Anisetti.

Digital Object Identifier no. 10.1109/TCC.2022.3233421

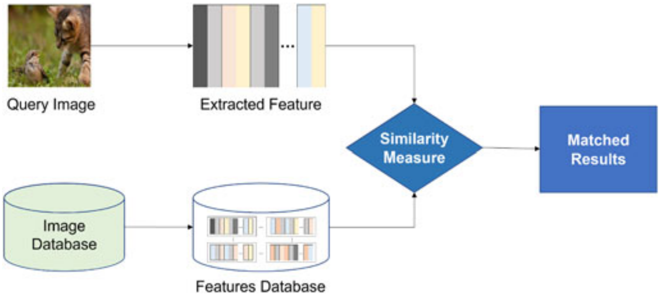


Fig. 1. The process of Content-based image retrieval(CBIR).

could be classified into two categories: feature-encryption based and image-encryption based PPCBIR schemes. In the first category of schemes, the image owner extracts feature vectors from their images at the beginning. Next, the images are protected by regular cryptographic tools and the image features are needed to be protected by elaborately-designed method for distance comparison. In the second kind of scheme, the owner needs only to encrypt the images. The rest of tasks, like feature extraction and image searching, are outsourced to the cloud, which further reduces the burden of image owners.

Contribution. This paper proposed a format-compatible JPEG PPCBIR scheme, which achieves good retrieval accuracy, high security, and no file expansion. The contributions can be summarized as follows:

1) *Outsourcing of feature extraction*. An format-compatible encryption method is designed to disturb the content of JPEG images, which includes three steps: binary code encryption, quantization table encryption, and big-block permutation. With such encrypted images, the features can be extracted directly by the cloud without interaction to the owner, which reduces the burden of image owners.
2) *No file expansion*. The proposed image encryption method is compatible to the JPEG format. It solves the file expansion problem caused by the encryption in the spatial domain.
3) *High retrieval accuracy*. With the specially-encrypted images, the local Markov features are calculated as local feature vectors and the BOW model is utilized to deal with these local features. Experimental results prove that our scheme performs better than the previous PPCBIR schemes in JPEG domain and is even comparable to the state-of-arts in spatial domain.

The rest of paper is arranged as follows. Related works are presented in Section 2. Preliminaries are introduced in Section 3. System overview is summarized in Section 4, and the detailed scheme is specified in Section 5. Sections 6 and 7 give the security analysis and experiments. In the end, the conclusions are summarized in Section 9.

2 Related Works

CBIR technologies are studied to retrieve similar images to a query from the large database of images by extracting and comparing certain elaborately-designed visual descriptor [4][5]. However, most of the existing CBIR methods are specialized for plaintext images, and such excellent techniques are generally unusable after the images are encrypted. Some efforts have been put to retrieve similar images with the privacy preserving. But both privacy and efficiency need improvement. Here we summarize the existing PPCBIR schemes into two classes.

Feature-Encryption Based Schemes. In this category of schemes, the owner extracts the features by himself. Next, images are protected by regular encryption tools, and features are protected with supporting the distance comparison in encryption domain. Lu et al. [6] made the first efforts on PPCBIR schemes. The authors generate the visual words by a vocabulary tree and characterize the image by the frequency histogram of visual features. The order-preserving encryption is employed to protect the histogram. In the same year, Lu et al. [7] provided three encryption technologies for the PPCBIR: bit plane randomization, random projection, and randomized unary encoding. These encryption steps can protect the features well but lead a decrease in retrieval accuracy. To get the comparable accuracy to that in plaintext domain [8], Lu et al. also suggested the homomorphic encryption for the protection of features. The homomorphic encryption is more secure and makes no loss on retrieval accuracy. However, the similarity calculation of homomorphic-encrypted features cause large computation and communication burdens. In [9], [10], Weng et al. proposed the PPCBIR schemes for large-scale image databases. Robust hash values are extracted and encrypted to represent images securely. The hamming distances of the encrypted hash values can be efficiently calculated. Besides, the authors tried to protect the search pattern by omitting some bits of feature in searching process. In [11], Xia et al. proposed a secure Earth Movers Distance (EMD) computation method and applied it to construct a PPCBIR scheme, but two-rounds of communications are involved. In [12], Xia et al. utilized the secure KNN to encrypt image features. Locality-sensitive hash is employed to speed up the searching and the watermarking method is designed to prevent illegal distribution in encryption domain. With dramatic development of Convolutional Neural Network (CNN), some researchers tried to construct the PPCBIR schemes by combining the CNN and feature protection methods. Hu et al. [13] designed a privacy region detection method to encrypt the sensitive part of an image. Then, CNN features and hash code are calculated and protected from the non-sensitive region for image searching. Zhang et al. [14] used the CNN to extract binary code and constructed a tree index to speed up the searching. Qin et al. [15] introduced the DenseNet network to extract features and employed the one-way hash to protect the features. Cheng et al. [16] aggregated VGG16 and VGG19 to extract features and utilized secure KNN to protect the features. Besides, secure CBIR schemes with multi-owner setting are also considered recently [17], [18], [19].

The feature-encryption-based schemes above are good solutions for secure outsourcing of image storage and CBIR services. However, in such methods, image owners have to extract features and build the index by themselves, which can be heavy burdens to the image owner. In [20], Yuan et al. attempted to let the cloud server build the tree index. However, the image owner in [20] is deeply involved during the index construction as the owner needs to perform decryption and encryption operations at each iteration when new cluster centers are generated.

Image-Encryption Based Schemes. This category of schemes can further relieve the burden of owner. In these schemes, the only task of image owner is to encrypt and upload the images. The key point is to build a lightweight encryption method that supports the feature extraction in encryption domain for the distance calculation. Bellafqira et al. [21] utilized homomorphic encryption to disturb image content and calculated SIFT features in encryption domain. However, such SIFT features are not usable for searching similar image. Xu et al. [22] used the partial encryption to construct an image retrieval scheme. The image is decomposed into two parts. One part is encrypted by Advanced Encryption Standard (AES) to keep the image secret while the other is kept unchanged for feature extraction and similarity comparison. Bernardo et al. [23] designed an image encryption method for the outsourcing of CBIR service. In this scheme, the pixels are substituted and permuted randomly to protect the image content. The histogram of encrypted pixel values are computed for similarity calculation. In [24], Xia et al. designed a PPCBIR scheme by using the BOW model. The authors protect the images by three encryption steps. Next, the local pixel histogram is extracted from each image block, which are then clustered together to produce the encrypted vocabularies. Finally, the frequency histograms of the visual words are utilized to retrieve images. Wang et al. [25] created a novel scheme which used AES and block permutation to protect images, random mapping features are extracted as visual words for searching. In [26], Xia et al. designed a PPCBIR scheme, which protect image content with random permutation and polyalphabetic cipher. Then, the Local Binary Pattern (LBP) descriptors are calculated for image retrieval.

Benefiting from the BOW model, the schemes [24], [25], [26] can achieve very good retrieval accuracy. However, the encryption conducted in the spatial domain destroys the correlation between the adjacent image pixels. Accordingly, the encrypted images cannot be compressed with a satisfying ratio.

JPEG-Compatible PPCBIR Schemes. There are many researches on Privacy-Preserving JPEG Image Retrieval. Zhang et al. [27] first employed the histogram invariance of DCT coefficients for images retrieval and achieve good retrieval performance. Nevertheless, the encrypted file expansion is large and the original image is partly visible, which brings higher storage cost and security threats. Cheng et al. [28] also proposed an PPCBIR scheme in DCT domain. The images are encrypted by shuffling the intermediate code of DCT coefficients, and the statistics of the code are calculated as features. After that, Zhang et al. [29] proposed an encrypted JPEG images retrieval scheme based on Markov model. The designed encryption algorithm ensured the security of image and also avoided file expansion. However, it needs to prepare image dataset for training in advance and fails to apply in unsupervised scenes. Furthermore, Liang et al. [30] found that the partial encryption exposed the feature of plaintext and thus is not secure for image retrieval. They encrypted the Huffman-code histogram to avoid feature explosion. However, their scheme didn't change the encrypted feature distribution, and thus unable to resist the background attack. In 2021, Li et al. [31] also proposed a privacy-preserving JPEG image retrieval scheme based on deep learning algorithms. However, they choose to encrypt the image globally and fails to preserve effective information for retrieval tasks.

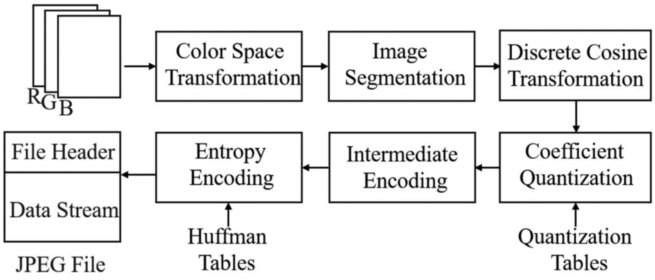


Fig. 2. JPEG compression process.

The key challenge of privacy-preserving JPEG image retrieval is how to design a balanced image protection algorithm, which can provide effective retrieval features while protecting image content. In this paper, we protect the image content by the VLI binary code encryption, quantization table encryption, and big-block permutation in a format-compatible manner. The use of the Markov model and BOW model guarantees the performance of image retrieval.

3 Preliminaries

3.1 JPEG Compression Process

JPEG is a widely used image format due to its high compression ratio and good image quality [32]. This paper proposes a format-compatible image encryption method. For easy understanding, we briefly introduced the JPEG compression process in Fig. 2, which is closely relevant to the structure of the JPEG file.

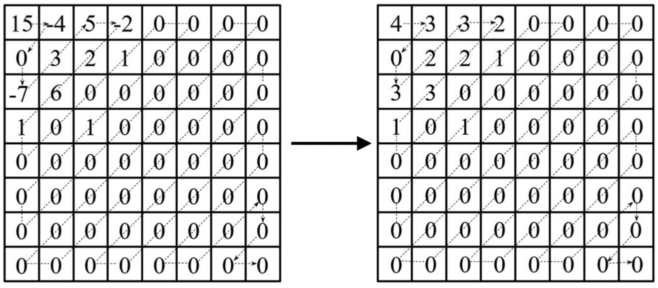
Step 1: Color Space Transformation. Images are generally displayed in RGB color space, but needs to be converted to YUV color space for JPEG compression, where Y denotes luminescence component, U and V denote chrominance components. Generally, we can put more compression on U and V components as human eyes are not so sensitive to them.

Step 2: Image Segmentation. After the color space transformation, the image is segmented into 8×8 blocks. Image compression is processed on these 8×8 blocks in YUV components.

Step 3: Discrete Cosine Transformation (DCT). In this step, DCT is performed on each 8×8 block. As a result, a 8×8 DCT coefficient block is generated, including 1 DC coefficient and 63 AC coefficients, which are denoted as c_{ij} where $i, j \in \{0, .., 7\}$.

Step 4: Quantization. The DCT coefficients are quantized to be integers as $c_{ij} \leftarrow \frac{c'_{ij}}{Q_{ij}}$, where Q_{ij}, $i, j \in \{0, .., 7\}$, are the elements in the quantization table Q. A JPEG image usually has two quantization tables, i.e., Q_Y for the Y component and Q_{UV} for the U and V components. The quantization tables are stored in JPEG files and can be different for each image. The encryption of quantization tables would protect image content to a large extent.

Step 5: Intermediate Encoding. In each 8×8 DCT coefficient block, the quantized coefficients are rearranged by



15	-4	5	-2	0	0	0	0
0	3	2	1	0	0	0	0
-7	6	0	0	0	0	0	0
1	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a)

4	3	3	2	0	0	0	0
0	2	2	1	0	0	0	0
3	3	0	0	0	0	0	0
1	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b)

Fig. 3. An example of group index matrix R, (a) an 8×8 DCT coefficient block in D, and (b) is the corresponding 8×8 block in R whose elements are replaced by the corresponding group indexes idx_G according to Table 1.

TABLE 1
Variable-Length Integer (VLI) Encoding Table

Value	Group index (idx_G)	Binary code (c_B)
0	0	-
-1,1	1	0,1
-3, -2, 2, 3	2	00,01,10,11
-7, ... ,-4, 4, ... , 7	3	000, ... , 111
-15 , ..., -8, 8, ..., 15	4	0000, ... , 1111
-32, ..., 16, 16, ..., 32	5	00000 , ... , 11111
-63, ..., -32, 32, ..., 63	6	000000 , ... , 111111
-127, ..., -64, 64, ..., 127	7	0000000 , ... , 1111111
-255, ..., -128, 128,...,255	8	00000000 , ... , 11111111
...	...	...
-32767, ..., -16384, 16384, ..., 32767	15	...

Zigzag scanning, generating a one-dimensional vector. For the DC coefficient, the difference between the current and former blocks are calculated, and encoded as $(null, v)$, where v denotes the difference. For AC coefficients, the non-zero coefficients are encoded to be the pairs of run-length and value (r, v), where v is the value of non-zero coefficient in the vector, and the run-length r is the number of consecutive zero-coefficients before the non-zero coefficient. Please note that r is definitely located in the range of $[0, \ldots 15]$ as the sequence of 16 consecutive zero will be encoded to be (15,0). As shown in Fig. 3a, the one-dimensional vector after Zigzag scanning is $(-15, -4, 0, -7, 3, 5, -2, 2, 6, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1)$. There are only 19 elements in the vector because there are 45 zero elements omitted after the last non-zero element. In addition, DC coefficients and AC coefficients are encoded in different ways. If the DC coefficient in former block is -18, the difference will be 3, which is represented as $(null, 3)$. Finally, the intermediate code of block can be calculated to be $\{(null, 3), (0, -4), (1, -7), (0, 3), (0, 5), (0, -2), (0, 2), (0, 6), \quad (0, 1), (3, 1), (4, 1), (0, 0)\}$.

Step 6: Entropy Encoding. The value v in (r, v) pair is then encoded to be a group index idx_G and a VLI binary code c_B according to the Variable-Length Integer (VLI) encoding table as presented in Table 1. Finally, the DCT coefficients are encoded as the triplets (r, idx_G, c_B).

In this way, the c_B is already the binary code. Encryption on c_B could cause a great disturbance to the image content. The first two elements, r and idx_G, are encoded into Huffman code. Finally, the JPEG image file is made up of these binary data and some other format information. Fig. 4 illustrates the structure of JPEG image files. SOI and EOI represent the beginning and the end of the file, respectively. APP section includes the information about image size, number of components, component sub-sampling, and so on. In this paper, we build big-blocks with the 8×8 blocks. We denote the big-blocks as $Bblk$ and extract local features from such these big-blocks.

3.2 Bag-of-Words Model

CBIR technologies retrieve similar images through comparing the similarity of the feature vectors extracted from the corresponding images. In general, the local features are more robust and can achieve better searching accuracy [8]. Bag-of-words (BOW) is a famous model that can use local features efficiently [24], [33], [34]. BOW model contains the following three steps:

Step 1: Local Features. In plaintext domain, typical local features includes SIFT [35], [36] and SURF [37], [38], which are often utilized in image retrieval. Some efforts have been put to calculate SIFT features in encryption domain. But the features in these schemes are the encrypted and incomputable ones and cannot be applied in image retrieval [39], [40], [41]. Xia et al. [24] calculated the color histograms as local features from small image blocks, which receive the improved retrieval accuracy than the global features. However, in [24], the images are encrypted in spatial domain and the resulting images cannot be compressed with a satisfying ratio, causing large storage expansion. In this paper, we extract Markov features from a big-block as the local feature in the JPEG compression process. It is worth noting that the local feature mentioned in BOW model is in fact a feature vector.

Step 2: Vocabulary. Local features can be multifarious. To obtain compact features, the local features are first calculated from the whole image dataset. Then all the local features are clustered to generate cluster centers which are regarded as the visual words to make up te vocabulary. The k-means clustering is a typical method for such work.

Step 3: Histogram of Visual Words. In the last step, each local feature is represented by its nearest cluster center. Then, an occurrence histogram of visual words are calculated for an image. This histogram has k bins and can be normalized to eliminate the effect of image size, resulting in a k-dimensional feature vector for similarity calculation.

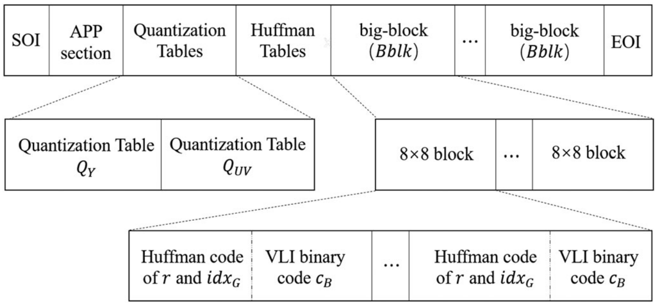


Fig. 4. The structure of JPEG image files.

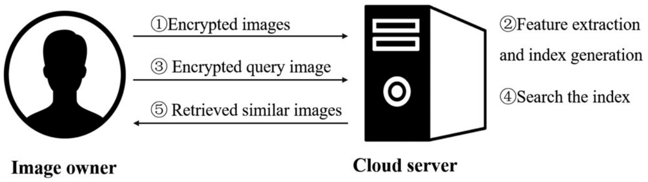


Fig. 5. System model.

3.3 Markov Model

Inspired by [29], we extract local features by Markov model from big-blocks which consist of multiple 8×8 DCT coefficient blocks. Markov model is a stochastic model for probabilistic systems [42], in which it is supposed that the future states only depend on the current states. This assumption simplifies the computation on some intractable problems. To further relieve the complexity, the hidden Markov model is proposed to only consider a part of current states, and the Markov transition probability matrix (MTPM) is defined to calculate the hidden Markov feature. The first order MTPM is defined as

$$\begin{aligned} M(x,y) &= p(S_t = y | S_{t-1} = x) \\ &= \frac{\sum \sigma(S_{t-1} = x, S_t = y)}{\sum \sigma(S_{t-1} = x)}, \end{aligned} \tag{1}$$

where S_t denotes the t-th state and $\sigma(\epsilon) = 1$ if ϵ holds, else $\sigma(\epsilon) = 0$. In this paper, we consider the sequence of DCT coefficients as the Markov chain, and similar images are assumed to hold similar Markov processes.

4 System Overview

4.1 System Model

Our scheme includes two kinds of entities: the image owner and cloud server, as illustrated in Fig. 5. At first, the image owner encrypts the images and uploads the encrypted images to the cloud server. Once receiving the ciphertext images, the cloud calculates features from them to provide the similar image retrieval service. If the owner needs to retrieve the similar images to a query, he encrypts the query image to be a secure trapdoor and sends it to the cloud. Once receiving the trapdoor, the cloud extracts features from it and search the similar ones by comparing the distances between the feature vector of query image and that of database images. Finally, similar images are return to the owner for decryption.

4.2 Treat Model

Similar to many searchable encryption schemes, we consider an semi-honest cloud server that correctly provides the storage and searching services but would be curious about the content of ciphertext images. The privacy leakage caused by access pattern [43], [44] are not considered in this paper. Besides, despite the cloud knows nothing about the image content, he knows the similarity among images. We name it as *similarity pattern* which is a inevitable trade-off for similar image retrieval.

4.3 Design Goals

Our scheme pursuits the following goals.

Accuracy. Our scheme is expected to achieve high retrieval accuracy. It means that the similar images should be included in search results and the dissimilar ones should be not. In this paper, the accuracy is ensured by the local Markov features and BOW model.

Security. The content of image should be protected from the cloud. But the privacy leaked by the access and similarity patterns is not discussed.

Relief on Owner's Burden. The proposed scheme is designed to outsource as many tasks as possible to the cloud server to relieve owner's burden. In the proposed scheme, the owners just need to encrypt and upload the image dataset.

5 The Proposed Scheme

In this section, a format compatible encryption method is designed for JPEG images at first, which cause no file size expansion. Next, Markov matrix is utilized to calculate local features from the chiphertext images. Finally, BOW model is applied to assemble the local features, generating a feature vector for each image. The similarities of images are compared according to the distances between the feature vectors.

5.1 Image Encryption

Images should be encrypted before outsourced for privacy preserving. There are three steps in our encryption method, i.e., binary code encryption, quantization table encryption, and big-block permutation. At first, we generate the secret keys for these encryption steps.

5.1.1 Key Generation

In initial phase, the image owner generates unique keys for different images so that the images are encrypted with one-time pad. The encryption steps contain four secret keys, including key_{c_B}, key_{Q_Y}, $key_{Q_{UV}}$ and pmt. The generation of secret keys is as follows:

$$\begin{cases} key_{c_B} \leftarrow \mathsf{PRF}(mk, ID || \text{``}c''_B), \\ key_{Q_Y} \leftarrow \mathsf{PRF}(mk, ID || \text{``}Q''_Y), \\ key_{Q_{UV}} \leftarrow \mathsf{PRF}(mk, ID || \text{``}Q''_{UV}), \\ pmt \leftarrow \mathsf{PRPG}(mk, ID, Blknum), \end{cases} \tag{2}$$

where PRF represents the pseudo-random number generation algorithm which generates random number sequences from the set of elements according to a random seed. The first parameter of PRF denotes the set of elements, and the second parameter denotes the random seed (a string), in which mk is the main secret key, ID denotes the image identity, and $||$ represents the concatenation operation of strings. $PRPG$ represents a pseudo-random permutation generator that produces a random permutation from a set of elements according to a random seed. $Blknum$ denotes the number of big-blocks and is also the length of the permutation sequences.

Our scheme ensures data privacy by using different secret keys for different JPEG images. Such operations seem to have destroyed the comparability of different images in terms of similarity. However, the run-length and group index of the JPEG image is not encrypted, which remains useful statistical information for image similarity comparison but keeps the image content unrevealed.

5.1.2 VLI Binary Code Encryption

The VLI binary code is extracted from the JPEG image file, and then concatenated to be a binary sequence, denoted as c_B. The binary code c_B is encrypted by xor operation as

$$c'_B \leftarrow c_B \oplus key_{c_B}. \tag{3}$$

With the encrypted c'_B, one cannot recover the original DCT coefficients. Specifically, the xor operation on VLI binary code does not change the bit length of the JPEG file.

5.1.3 Quantization Table Encryption

As mentioned in Section 3.1, there are two quantization tables Q_Y and Q_{UV}, which are essential to reconstruct images from JPEG files. In addition, images can hold different quantization tables to each other. Thus, the encryption of quantization tables provides additional protection to the image content. Formally, the quantization tables are encrypted as follows"

$$\begin{cases} Q'_Y \leftarrow Q_Y \oplus key_{Q_Y}, \\ Q'_{UV} \leftarrow Q_{UV} \oplus key_{Q_{UV}}. \end{cases} \tag{4}$$

5.1.4 Big-Block Permutation

Images are divided into 8 × 8 blocks for JPEG compression. To protect the image content and support feature extraction, we divide the image into big-blocks as illustrated in Fig. 6 for permutation and local feature extraction. Each big-block consists of multiple adjacent DCT blocks. Denoting the i-th big-block in an image as Blk_i, the big-block permutation is executed as

$$Blk'_i \quad \leftarrow \quad Blk_{pmt[i]} \quad , \tag{5}$$

where $i = 1, .., Blknum$, and $Blknum$ is the amount of big-blocks in an image. Please note that, due to down-sampling in JPEG compression, the Y component is of twice size to the U and V components in both the width and height. Accordingly, we set the size of big-blocks in Y component twice to that in U and V components in height and width. As a result, the YUV components hold the same number of big-blocks. The big-block permutation makes no change to the file size of the image.

5.2 Feature Extraction

After receiving the ciphertext images, the cloud should be able to calculate useful features from the ciphertext images without interaction to owners. Here, we divide the feature extraction process into four steps. First, the cloud preprocesses the image data for feature extraction. Second, local Markov features are calculated by the data in big-blocks. Third, the k-means algorithm is employed to construct vocabulary with the local Markov features. Finally, the feature vector are generated to measure the similarity between images.

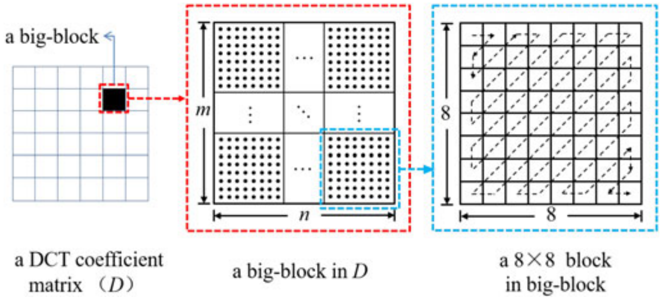


Fig. 6. The construction of big-block

TABLE 2
The Occurrence Probability of Different idx_G, Which is Calculated From Inria Holidays Database

Group index (idx_G)	occurrence (%)
0	54.0466
1	20.3431
2	7.0755
3	10.4306
4	3.9264
5	2.3180
6	1.1935
7	0.4898
8	0.1465
9	0.0286
10	0.0014
11	0
12	0
13	0
14	0
15	0

5.2.1 Data Pre-Processing

In the proposed encryption method, the VLI code, quantization table, and the order of big-blocks are encrypted, but the run-length r and group index idx_G are unchanged. In addition, the positions of big-blocks are shuffled while the positions of 8 × 8 DCT coefficient blocks in a big-block are kept unchanged. These encryption steps can protect the images and leave some information for the similar image search.

Matrix of Group Index. With the encrypted binary code c_B, the unencrypted run-length r and group index idx_G, one can construct a matrix of encrypted DCT coefficients. However, this matrix (denoted as D) is useless for similar image retrieval. However, the group index idx_G are unchanged in image encryption process. With these data, we can reconstruct a matrix R with the same dimension to the DCT coefficient matrix D. Every element in R is equal to the corresponding group index idx_G of the element in D as shown in Fig. 3. In addition, since the group index of zero DCT coefficient is zero (see Table 1), the zeroes in D are still zeroes in R.

Data Truncation. As shown in Table 1, the group indexes idx_G have a fixed value range [0,15]. The occurrence probability of large idx_G is quite low. As listed in Table 2, most idx_G locates in [0,...,6]. Thus, we can shrink the group indexes idx_G to a small range to reduce the feature dimension as

$$idx_G = \begin{cases} idx_G, & if\ idx_G <= \tau, \\ \tau, & if\ idx_G > \tau. \end{cases} \tag{6}$$

where τ is experimental value.

5.2.2 Local Markov Feature Generation

This subsection presents the feature extraction from the group index matrix R. The group index matrices of YUV components are denoted as R_Y, R_U, and R_V. It is worthy noting that the size of R_U and R_V are half that of R_Y in both the height and width. The matrices R are divided into big-blocks Blk in the same way as that in image encryption. Denotes the big-blocks from YUV components as Blk_Y, Blk_U, and Blk_V. Accordingly, the size of Blk_Y is twice of Blk_U and Blk_V in both the height and width. Three kinds of local Markov features are calculated from each big-block, including intra-block, inter-block, and inter-component features.

Intra-Block Features. Intra-block features characterize the dependency of adjacent DCT coefficients. Denote 8×8 group index block as blk whose elements, $blk[i], i = 0, \ldots, 63$, are rearranged by Zig-zag scanning as illustrated in Fig. 3b. Then, a Markov transition probability matrix (MTPM) M in the 8×8 block is calculated as

$$\begin{aligned} M(x,y) &= p(S_t = y|S_{t-1} = x) \\ &= \frac{\sum_{i=1}^{63} \sigma(blk[i-1] = x, blk[i] = y)}{\sum \sigma(blk[i-1] = x)}, \end{aligned} \tag{7}$$

where $x, y \in [0, \tau]$, and $\sigma(\epsilon) = 1$ if ϵ holds, else $\sigma(\epsilon) = 0$.

Next, an averaged MTPM, denoted as $M_{intrablock}$, is calculated from all 8×8 blocks in a big-block as

$$M_{intrablock}(x,y) = \frac{\sum_{i=k}^{blknum} M_k(x,y)}{blknum}, \tag{8}$$

where M_k denotes the MTPM calculated from the k-th 8×8 block in a big-block, and $blknum$ is the amount of 8×8 blocks in the big-block. The elements of $M_{intrablock}$ are used as the intra-block Markov features. Since $x, y \in [0, \tau]$, the dimension of intra-block Markov features equals to $3 \times (1 + \tau) \times (1 + \tau)$ for the three components.

Inter-Block Features. Inter-block features characterize the dependency of DCT coefficients between the adjacent 8×8 blocks. The MTPMs are calculated from each big-block horizontally, vertically, diagonally, and mirror-diagonally. We present the calculation of MTPM in a big-block as

$$\begin{cases} M_H(x,y) = p(S_t = y|S_{t-1} = x) \\ = \frac{\sum_{i=0}^{m'} \sum_{j=0}^{n'-1} \sum_{t=0}^{63} \sigma(blk_{i,j}[t]=x, blk_{i,j+1}[t]=y)}{\sum_{i=0}^{m'} \sum_{j=0}^{n'-1} \sum_{t=0}^{63} \sigma(blk_{i,j}[t]=x)}, \\ M_V(x,y) = p(S_t = y|S_{t-1} = x) \\ = \frac{\sum_{i=0}^{m'-1} \sum_{j=0}^{n'} \sum_{t=0}^{63} \sigma(blk_{i,j}[t]=x, blk_{i+1,j}[t]=y)}{\sum_{i=0}^{m'-1} \sum_{j=0}^{n'} \sum_{t=0}^{63} \sigma(blk_{i,j}[t]=x)}, \\ M_D(x,y) = p(S_t = y|S_{t-1} = x) \\ = \frac{\sum_{i=0}^{m'-1} \sum_{j=0}^{n'-1} \sum_{t=0}^{63} \sigma(blk_{i,j}[t]=x, blk_{i+1,j+1}[t]=y)}{\sum_{i=0}^{m'-1} \sum_{j=0}^{n'-1} \sum_{t=0}^{63} \sigma(blk_{i,j}[t]=x)}, \\ M_M(x,y) = p(S_t = y|S_{t-1} = x) \\ = \frac{\sum_{i=0}^{m'-1} \sum_{j=1}^{n'} \sum_{t=0}^{63} \sigma(blk_{i,j}[t]=x, blk_{i+1,j-1}[t]=y)}{\sum_{i=0}^{m'-1} \sum_{j=1}^{n'} \sum_{t=0}^{63} \sigma(blk_{i,j}[t]=x)}, \end{cases} \tag{9}$$

where $m' = m/8 - 1$, $n' = n/8 - 1$, and m, n are the height and width of the big-block Blk. The elements of these MTPMs are used as inter-block Markov features. The inter-block Markov features are calculated from three components separately. Thus, the dimension of inter-block Markov features equals to $3 \times 4 \times (1 + \tau) \times (1 + \tau)$ for the three components.

Inter-Component Features. Inter-component features characterize the dependency of DCT coefficients among the three color components. Calculate the markov probability transfer matrixes between three components, generating three MTPMs M_{YU}, M_{YV} and M_{UV}. In addition, due to the down-sampling mode in the JPEG process, the height and width of the images in Y component are twice those in U and V components. Accordingly, the big-blocks in Y component have four times elements two the big-blocks in U and V components.

We divide the big-blocks in Y component into four parts denoted as Blk_{Y_1}, Blk_{Y_2}, Blk_{Y_3} and Blk_{Y_4}, which are with the same size of Blk_U and Blk_V. The MTPMs M_{YU} is calculated as

$$\begin{cases} M_{Y_1U}(x,y) = p(S_t = y|S_{t-1} = x) \\ = \frac{\sum_{i=1}^{m/2} \sum_{j=1}^{n/2} \sigma(Blk_{Y_1}[i,j]=x, Blk_U[i,j]=y)}{\sum_{i=1}^{m/2} \sum_{j=1}^{n/2} \sigma(Blk_{Y_1}[i,j]=x)}, \\ M_{Y_2U}(x,y) = p(S_t = y|S_{t-1} = x) \\ = \frac{\sum_{i=1}^{m/2} \sum_{j=1}^{n/2} \sigma(Blk_{Y_2}[i,j]=x, Blk_U[i,j]=y)}{\sum_{i=1}^{m/2} \sum_{j=1}^{n/2} \sigma(Blk_{Y_2}[i,j]=x)}, \\ M_{Y_3U}(x,y) = p(S_t = y|S_{t-1} = x) \\ = \frac{\sum_{i=1}^{m/2} \sum_{j=1}^{n/2} \sigma(Blk_{Y_3}[i,j]=x, Blk_U[i,j]=y)}{\sum_{i=1}^{m/2} \sum_{j=1}^{n/2} \sigma(Blk_{Y_3}[i,j]=x)}, \\ M_{Y_4U}(x,y) = p(S_t = y|S_{t-1} = x) \\ = \frac{\sum_{i=1}^{m/2} \sum_{j=1}^{n/2} \sigma(Blk_{Y_4}[i,j]=x, Blk_U[i,j]=y)}{\sum_{i=1}^{m/2} \sum_{j=1}^{n/2} \sigma(Blk_{Y_4}[i,j]=x)}, \\ M_{YU} = (M_{Y_1U} + M_{Y_2U} + M_{Y_3U} + M_{Y_4U})/4 \end{cases} \tag{10}$$

where m, n are the height and width of the big-block Blk_Y. The MTPM M_{YV} is calculated in the same way as M_{YU}, and the MTPM M_{UV} is calculated as

$$\begin{cases} M_{UV}(x,y) = p(S_t = y|S_{t-1} = x) \\ = \frac{\sum_{i=1}^{m/2} \sum_{j=1}^{n/2} \sigma(Blk_U[i,j]=x, Blk_V[i,j]=y)}{\sum_{i=1}^{m/2} \sum_{j=1}^{n/2} \sigma(Blk_U[i,j]=x)}, \end{cases} \tag{11}$$

The dimension of inter-component Markov features is $3 \times (1 + \tau) \times (1 + \tau)$.

Summation of Feature Extraction. In our scheme, the Markov model is used to extract local features including intra-block features, inter-block features, and inter-component features, which build up the local feature vector with $18 \times (1 + \tau) \times (1 + \tau)$ elements.

5.2.3 Vocabulary Generation

Through the steps above, the image owner can extract a great number of local Markov features from the encrypted image dataset. These local features (vectors) are clustered by k-means algorithm, resulting k cluster centers. The cluster centers are rightly the visual words which build up the vocabulary.

5.2.4 Histogram of Visual Words

For an image, we can calculate a batch of local Markov features from its big-blocks. In our scheme, each local Markov feature will count one to its nearest visual word. Then, a histogram of visual words can be generated for each image and will be normalized by the amount of big-blocks in the image. The image similarities are then compared according to the distances between the normalized histogram.

5.3 Image Search

To retrieve similar images, the owner encrypts the query image according to the three steps in Section 5.1, and then uploads the encrypted query to cloud. Then, the cloud generates a feature vector from the ciphertext query according to the encryption steps in Section 5.2. Next, the Manhattan distance between the feature vector is calculated as

$$d(q, f) = \sum_{i=1}^{k} |q[i] - f[i]|, \tag{12}$$

where q and f represent the normalized histograms calculated from the query and the database image. After the calculation, the image owner will get the similar images returned from the cloud server. After the decryption, the plaintext image can be obtained.

6 Security Analysis

An PPCBIR scheme generally considers an semi-honest cloud server, which will correctly provide the storage and similar image searching services but may be interested in the content of ciphertext images. The cloud is considered as the only adversary in this scheme. As presented in Section 5.1.1, An unique key is generated for each image. Therefore, the adversary can only execute a brute-force attack on the encrypted image. The security is discussed under the ciphertext-only attack (COA) model.

Summary of Information Leakage. In order to support the similarity comparison between encrypted images, some information about the image is leaked to the server, including the size of big-block, the number of big-block $Blknum$, the length of binary code c_B, the size of quantization table, and the permuted group index matrices.

Theorem 1. *With an honest-but-curious cloud server under the COA model, the ciphertext images are computationally secure. The security strength sec_{stren} is equal to*

$$sec_{stren} = len_{c_B} + len_{Q_Y} + len_{Q_{UV}} + \log_2^{Blknum!}, \tag{13}$$

where len_{c_B} is the total length of binary code, len_{Q_Y} and $len_{Q_{UV}}$ are the sizes of two quantization tables respectively, and $Blknum$ is the amount of big-blocks in the image.

Proof. As unique keys are generated for different images, the adversary can only conduct brute-force attack on the encrypted image. Then, the scheme can be computationally secure if the defined key space is large enough. The security strength equals to the length of secret key. First, the VLI binary code c_B is encrypted by xor operation. The key length is equal to the total length of c_B. Second, two quantization tables are also encrypted by xor operation. The key length is equal to the sum of length of Q_Y and Q_{UV}. Finally, the big-blocks are encrypted by permutation. The computation complexity to recover the original order equals to $\log_2^{Blknum!}$. The total security strength is the combination of three steps. When the image size is 64×64, len_{c_B}, len_{Q_Y}, $len_{Q_{UV}}$, and $Blknum$ equals 5504, 512, 512, and 64. Then, we get a 6824 bits key space which can be computationally secure under current computation power. □

7 Experimental Results

The performance of the proposed scheme is tested in terms of visual effectiveness of encryption, time consumption, and retrieval accuracy. We implemented our scheme by Matlab 2019b and tested it with the Intel Core (TM) i7-6900K (3.20 GHz). Inria Holidays image set [45] is used for testing. It includes 1491 color images with the size 3264×2448 or 2448×3264. The 1491 color images are divided into 500 categories, In each of which, the first image is used as the query and the rests are considered as its similar images. There are a Python evaluation package for mAP calculation at the homepage of dataset. It simplifies the fair comparison of retrieval accuracy. In our experiment, we set the parameter in Data truncation $\tau = 8$. As shown in Table 2, when we consider the idx_G in [0,8], 99.97% idx_G are covered, but the dimension of local feature vector decrease from 6498 to 1458. Thus, we believe 8 is a suitable value for τ, which can preserve enough information while reducing feature dimension.

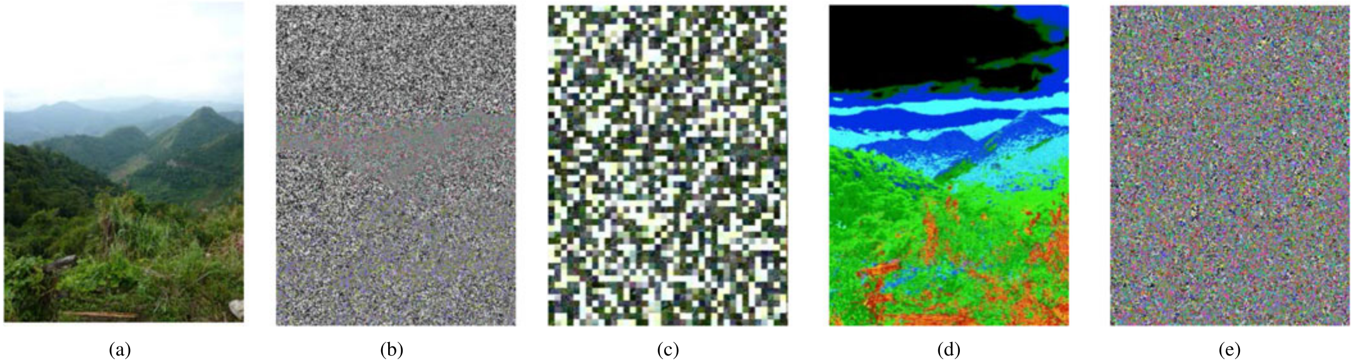


Fig. 7. Effectiveness of image encryption, (a) the original image, (b) the image with VLI binary code encrypted, (c) the image with big-blocks permuted, (d) the image with quantization tables encrypted, and (e) the image encrypted by the three encryption steps.

TABLE 3
The Time Cost on Image Encryption (s)

Methods	Time Consumption(s)			
Our scheme	VLI encryption 2.6728	Quantization table encryption 0.0001	Big-block permutation 0.2221	Total 3.0862
BOEW[24]	Block Permute 0.1115	Permute In Block 1.0491	Value Permute 2.7357	Total 4.1965
IES-CBIR [23]		-		**1.0884**
Cheng [29]		-		1.8057

TABLE 4
The Time Cost on Feature Extraction (s)

Methods	Time Consumption under different $Blksize$					
	32	64	96	128	160	240
BOEW [24]	0.2767	0.23	0.2213	0.227	0.2246	0.2186
Our scheme	6.0631	2.3473	1.055	0.8443	0.4925	0.3262
MIPP [46]	0.2969					

7.1 Effectiveness of Image Encryption

In the proposed scheme, images are protected by binary code encryption, quantization table encryption, and big-block permutation. Here, we present the separate and joint effectiveness of the three protection steps. As illustrated in Fig. 7, the quantization table encryption can only disturb the color information to some extent, while the binary code encryption and big-block permutation can provide a protection on the image content. In total, the image content is well disturbed by combining three encryption steps as shown in Fig. 7e.

7.2 Time Cost

This section tests the time cost of image encryption, feature extraction, and search. The time cost of three encryption steps are listed in Table 3, which are averaged from the encryption of 1491 images. Our scheme shows lower time consumption of image encryption than the BOEW [24]. Compared with IES-CBIR [23] and Cheng [29], our scheme shows a higher encryption overhead. However, the encryption methods are related to the feature extraction process. The IES-CBIR and Cheng use global features to analyze the preserved information, and fails to perform accurate image retrieval. Differently, our scheme preserves local information during image encryption process, and extract the local features to perform image retrieval. The experimental results in Table VII prove our effectiveness.

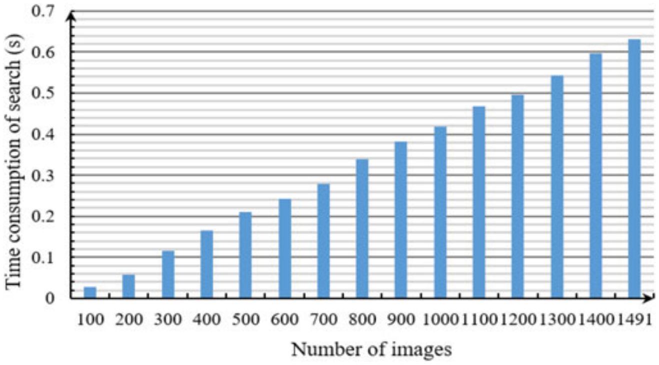


Fig. 8. The time cost of search with different numbers of images ($k = 8000$)

The time cost of feature extraction is listed in Table 4, which are also averaged from 1491 images. The feature extraction process is time-consuming but important for accurate retrieval. From 4, the time for feature extraction is associated with the size of big-blocks. A larger $Blksize$ means fewer big-blocks, and thus takes less time in feature extraction.

In our scheme, we only construct a one-to-one index. Therefore, the time cost of search is linear to the image amount in dataset, as shown in and Fig. 8. We also compared the searching time with BOEW [24] in Table 5. Our scheme adopts a linear search strategy for searching at the cloud server, and the search efficiency depends on cluster number k. The search times with different cluster number k are close to BOEW.

7.3 Search Accuracy

The mean Average Precision (mAP) is calculated to evaluate the search accuracy. The python evaluation package in [45] is utilized to calculate mAP of schemes for fair comparison. The search accuracy of our scheme is associated to the size of big-block and the number of cluster centers. But it can be found in Table 6 that our scheme is not so sensitive to the parameters. When we choose the $Blksize$ from 64×64 to

TABLE 5
The Time Cost of Search With Different k (s)

Methods	k					
	100	500	1000	3000	5000	8000
BOEW [24]	0.0086	0.0284	0.0710	0.2204	0.3289	0.5438
Ours	0.0088	0.0286	0.0612	0.2100	0.3462	0.5384

TABLE 6
mAP Value of Our Proposed Scheme

k	32×32	64×64	96×96	128×128	160×160
500	0.55820	0.57151	0.56446	0.54456	0.52349
1000	0.57747	0.58094	0.58636	0.57656	0.56410
3000	0.59506	0.62315	0.61752	0.60649	0.59919
5000	0.59760	0.61871	0.61934	0.61430	0.59979
8000	0.60095	**0.63205**	0.63021	0.61862	0.61440
15000	0.60665	0.62513	0.62037	0.60618	0.60590
Average	0.58932	0.60858	0.60637	0.59445	0.58448

128 × 128 and the k from 3000 to 15000, we can always obtain a mAP larger than 0.6, and the highest mAP is achieved when $Bblsize$ is set as 64×64 and k equals to 8000, which outperforms many state-of-art PPCBIR schemes. As shown in Table 7, the schemes in spatial domain generally get better retrieval accuracy than that for JPEG images as the pixel values are more informative than the quantized AC coefficients. Our scheme outperforms two previous JPEG-domain schemes [28], [29] and is comparable to the state-of-the-arts [22], [23], [24] in spatial domain. We attribute the search accuracy to the utilization of the format-compatible encryption method and the feature extraction process. The extracted local features tend to have better retrieval performance than global features, which has similar results with [47], [48]. The block encryption strategy could well preserve local statistical information when encrypting images. Also, the used Markov model and Bag-Of-Words (BOW) model could extract the preserved local feature from the encrypted image, and organize local features accurately. Thus, our scheme could perform high-precision image retrieval.

7.4 Expansion of File Size

The three encryption steps in our scheme are compatible to JPEG compression process and make no influence on the size of encrypted files. Specifically, the xor operation on VLI binary code and quantization tables do not change the bit length. In addition, the big-block permutation also makes no change to the file size of image. We compared the size of the encrypted image database (Inria Holidays) with that of previous scheme, and the results are shown in Table 8.

7.5 Query Unlinkability

In image retrieval tasks, the server could match the search query and encrypted image always when the query is deterministic. In this subsection, we evaluate the query unlinkability of our scheme To prevent server from simply matching deterministic features. We add some noise into the query image so as to make slight changes to the search result as [49]. Gaussian noises with different stand deviations are added into query images to disturb image contents. As shown in Fig. 9, the mean Average Precision (mAP) is calculated to show the search accuracy with different noise standard deviations. It can be seen that when the stand deviation is less than 15, the mAP remains around 0.63. Even though the query images and features are disturbed, our scheme could maintain search accuracy as long as the image content is not severely damaged.

TABLE 7
The Comparison of mAPs With Previous Schemes

Schemes	Performed in JPEG or spatial domain	mAP
Partial-encryption [22]	In spatial	0.56040
IES-CBIR [23]	In spatial	0.54564
BOEW [24]	In spatial	0.62641
Cheng et al. [28]	For JPEG	0.36000
Cheng et al. [29]	For JPEG	0.54187
Li et al. [31]	For JPEG	0.4754
Our Scheme	For JPEG	**0.63205**

TABLE 8
File Extension Comparison With Classic Schemes

The classic schemes	The file size (GB)
Our proposed scheme	**2.65**
IES-CBIR[23]	20.02
BOEW[24]	17.97

8 Future Works

This paper proposes a secure JPEG-format-compatible image retrieval scheme. Markov features are designed for high retrieval accuracy. In the future, it could be worth improving the scheme in three aspects: 1) About retrieval accuracy. Deep learning has achieved great success in plaintext image retrieval. It could be a promising way to learn effective features by deep learning techniques for better retrieval accuracy. 2) About security. Our scheme leaves the group index and run-length unencrypted for image retrieval. It is preferred to get all data encrypted, but there is a trade-off between security and search accuracy. Also, feature encryption would increase the security of image retrieval, but there is a tradeoff between feature encryption and retrieval efficiency. 3) About efficiency. To extract effective features from such encrypted images, we inevitably perform high-complexity feature extraction rather than simple statistical analysis. The time consumption is acceptable for user side, but there is still much room for improvement. At the same time, our scheme just considers a linear index which is not an efficient one. In fact, some existing index structures [50], [51] can be directly applied to improve the efficiency of our scheme.

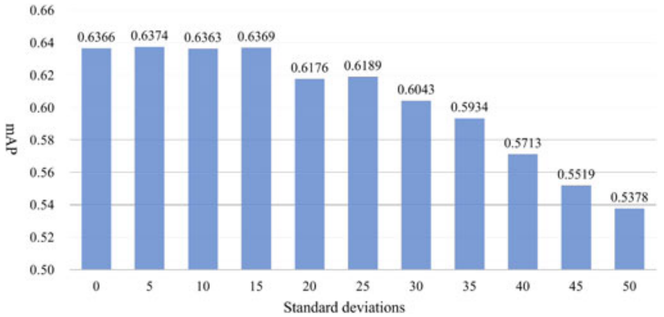


Fig. 9. The mean Average Precision (mAP) with different noise levels. The abscissa represents the standard deviation of added noise.

9 Conclusion

This paper proposed a PPCBIR scheme for JPEG images. A format-compatible encryption method with three steps is designed to protect the content of images without expansion of file size. Local Markov features are extracted directly from the encrypted image file for the search of similar image. BOW model is utilized to exploit the local features to achieve good retrieval accuracy. In the proposed scheme, the image storage, feature extraction, and image searching are outsourced to the server, which reduces the burden of owner. In future, it would be worth designing better feature extraction methods to improve search accuracy.

References

[1] S. Çiftçi, A. O. Akyüz, and T. Ebrahimi, "A reliable and reversible image privacy protection based on false colors," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 68–81, Jan. 2018.

[2] I. Scott, "Instagram breach exposes personal data of 49 million users," [Online]. Available: https://www.cpomagazine.com/cyber-security/instagram-breach-exposes-personal-data-of-49-million-users/

[3] S. Adario, "Jennifer lawrence and other celebs hacked as nude photos circulate on the web," [Online]. Available: https://www.mashable.com/2014/08/31/celebrity-nude-photo-hack/

[4] I. González-Díaz, C. E. Baz-Hormigos, and F. Díaz-de-María, "A generative model for concurrent image retrieval and roi segmentation," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 169–183, Jan. 2014.

[5] L. Dong, Y. Liang, G. Kong, Q. Zhang, X. Cao, and E. Izquierdo, "Holons visual representation for image retrieval," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 714–725, Apr. 2016.

[6] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, "Enabling search over encrypted multimedia databases," in *Media Forensics Secur.*, 2009, Art. no. 725418.

[7] W. Lu, A. L. Varna, A. Swaminathan, and W. Min, "Secure image retrieval through feature protection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2009, pp. 1533–1536.

[8] W. Lu, A. L. Varna, and W. Min, "Confidentiality-preserving image search: A comparative study between homomorphic encryption and distance-preserving randomization," *IEEE Access*, vol. 2, pp. 125–141, 2014.

[9] L. Weng, L. Amsaleg, A. Morton, and S. Marchandmaillet, "A privacy-preserving framework for large-scale content-based information retrieval," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 152–167, Jan. 2015.

[10] L. Weng, L. Amsaleg, and T. Furon, "Privacy-preserving outsourced media search," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 10, pp. 2738–2751, Oct. 2016.

[11] Z. Xia, Y. Zhu, X. Sun, Z. Qin, and K. Ren, "Towards privacy-preserving content-based image retrieval in cloud computing," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 276–286, Jan.–Mar. 2018.

[12] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2594–2608, Nov. 2016.

[13] L. Hu, T. Xiang, and S. Guo, "Sensir: Towards privacy-sensitive image retrieval in the cloud," *Signal Process.: Image Commun.*, vol. 84, 2020, Art. no. 115837.

[14] C. Zhang, L. Zhu, S. Zhang, and W. Yu, "TDHPPIR: An efficient deep hashing based privacy-preserving image retrieval method," *Neurocomputing*, vol. 406, pp. 386–398, 2020.

[15] Z. Wang, J. Qin, X. Xiang, and Y. Tan, "A privacy-preserving and traitor tracking content-based image retrieval scheme in cloud computing," *Multimedia Syst.*, vol. 27, pp. 403–415, 2021.

[16] S.-L. Cheng, L.-J. Wang, G. Huang, and A.-Y. Du, "A privacy-preserving image retrieval scheme based secure KNN, DNA coding and deep hashing," *Multimedia Tools Appl.*, vol. 80, no. 15, pp. 22 733–22 755, 2021.

[17] M. Shen, G. Cheng, L. Zhu, X. Du, and J. Hu, "Content-based multi-source encrypted image retrieval in clouds with privacy preservation," *Future Gener. Comput. Syst.*, vol. 109, pp. 621–632, 2020.

[18] Q. Gu, Z. Xia, and X. Sun, "Msppir: Multi-source privacy-preserving image retrieval in cloud computing," 2020, *arXiv:2007.12416*.

[19] Y. Li et al. "Traceable and controllable encrypted cloud image search in multi-user settings," *IEEE Trans. Cloud Comput.*, vol. 10, no. 4, pp. 2936–2948, Oct.–Dec. 2022.

[20] J. Yuan, S. Yu, and L. Guo, "SEISA: Secure and efficient encrypted image search with access control," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 2083–2091.

[21] R. Bellafqira, G. Coatrieux, D. Bouslimi, and G. Quellec, "An end to end secure CBIR over encrypted medical database," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.* , 2016, pp. 2537–2540.

[22] Y. Xu, J. Gong, L. Xiong, Z. Xu, J. Wang, and Y.-q. Shi, "A privacy-preserving content-based image retrieval method in cloud environment," *J. Vis. Commun. Image Representation*, vol. 43, pp. 164–172, 2017.

[23] B. Ferreira, J. Rodrigues, J. LeitÃ£o, and H. Domingos, "Practical privacy-preserving content-based retrieval in cloud image repositories," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 784–798, Jul.–Sep. 2019.

[24] Z. Xia, L. Jiang, D. Liu, L. Lu, and B. Jeon, "BOEW: A content-based image retrieval scheme using bag-of-encrypted-words in cloud computing," *IEEE Trans. Services Comput.*, vol. 15, no. 1, pp. 202–214, Jan./Feb. 2019.

[25] H. Wang, Z. Xia, J. Fei, D. Liu, L. Lu, and B. Jeon, "An AES-based secure image retrieval scheme using random mapping and BOW in cloud computing," *IEEE Access*, vol. 8, pp. 61138–61147, 2020.

[26] Z. Xia, L. Wang, J. Tang, N. Xiong, and J. Weng, "A privacy-preserving image retrieval scheme using secure local binary pattern in cloud computing," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 318–330, First Quarter 2021.

[27] X. Zhang and H. Cheng, "Histogram-based retrieval for encrypted JPEG images," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process.*, 2014, pp. 446–449.

[28] H. Cheng, X. Zhang, J. Yu, and Y. Zhang, "Encrypted JPEG image retrieval using block-wise feature comparison," *J. Vis. Commun. Image Representation*, vol. 40, pp. 111–117, 2016.

[29] H. Cheng, X. Zhang, J. Yu, and F. Li, "Markov process-based retrieval for encrypted JPEG images," *EURASIP J. Inf. Secur.*, vol. 2016, no. 1, 2016, Art. no. 1.

[30] H. Liang, X. Zhang, and H. Cheng, "Huffman-code based retrieval for encrypted JPEG images," *J. Vis. Commun. Image Representation*, vol. 61, pp. 149–156, 2019.

[31] B. Li, S. Ding, and X. Yang, "A privacy-preserving scheme for JPEG image retrieval based on deep learning," in *J. Phys.: Conf. Ser.*, 2021, Art. no. 012007.

[32] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. xviii–xxxiv, Feb. 1992.

[33] W. Zhong, Q. Ke, S. Jian, and H. Y. Shum, "A multi-sample, multi-tree approach to bag-of-words image representation for image retrieval," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 1992–1999.

[34] J. Wang et al."Bag-of-features based medical image retrieval via multiple assignment and visual words weighting," *IEEE Trans. Med. Imag.*, vol. 30, no. 11, pp. 1996–2011, Nov. 2011.

[35] E. N. Mortensen, H. Deng, and L. G. Shapiro, "A SIFT descriptor with global context," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 184–190.

[36] C. Kang, L. Zhu, X. Qian, J. Han, M. Wang, and Y. Y. Tang, "Geometry and topology preserving hashing for sift feature," *IEEE Trans. Multimedia*, vol. 21, no. 6, pp. 1563–1576, Jun. 2019.

[37] J. R. R. Uijlings, A. W. M. Smeulders, and R. J. H. Scha, "Real-time visual concept classification," *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 665–681, Nov. 2010.

[38] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[39] C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, "Image feature extraction in encrypted domain with privacy-preserving SIFT," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4593–4607, Nov. 2012.

[40] Z. Qin, J. Yan, and K. Ren, "Private image computation: The case of cloud based privacy-preserving SIFT," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2014, pp. 179–180.

[41] Q. Zhan, J. Yan, K. Ren, W. C. Chang, and W. Cong, "SecSIFT: Secure image sift feature extraction in cloud computing," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 4, pp. 1–24, 2016.

[42] P. A. Gagniuc, *Markov Chains: From Theory to Implementation and Experimentation*. Hoboken, NJ, USA: Wiley, 2017.

[43] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data," in *Proc. IEEE 28th Int. Conf. Data Eng.*, 2012, pp. 1156–1167.

[44] T. Hoang, A. A. Yavuz, and J. G. Merchan, "A secure searchable encryption framework for privacy-critical cloud storage services," *IEEE Trans. Services Comput.*, vol. 14, no. 6, pp. 1675–1689, Nov./Dec. 2021.

[45] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 304–317.

[46] M. Shen, G. Cheng, L. Zhu, X. Du, and J. Hu, "Content-based multi-source encrypted image retrieval in clouds with privacy preservation," *Future Gener. Comput. Syst.*, vol. 109, pp. 621–632, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17321969

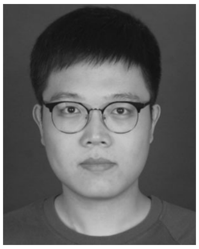
[47] X.-Y. Wang, J.-F. Wu, and H.-Y. Yang, "Robust image retrieval based on color histogram of local feature regions," *Multimedia Tools Appl.*, vol. 49, no. 2, pp. 323–345, Aug. 2010.

[48] C. Celik and H. S. Bilge, "Content based image retrieval with sparse representations and local feature descriptors : A comparative study," *Pattern Recognit.*, vol. 68, pp. 1–13, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317301048

[49] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 8, pp. 1467–1479, Aug. 2011.

[50] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. 20th Annu. Symp. Comput. Geometry*, 2004, pp. 253–262.

[51] T. Zhang, G.-J. Qi, J. Tang, and J. Wang, "Sparse composite quantization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4548–4556.

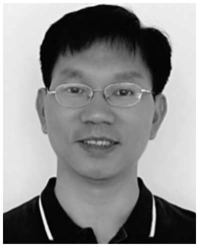


Peipeng Yu received the master's degree from the Nanjing University of Information Science and Technology, in 2022. He is currently working toward the PhD degree with Jinan University. His research interests include artificial intelligence security and video forensics.

Jian Tang received the BS degree from Heilongjiang Bayi Agricultural University, in 2018, and the MS degree from the College of Computer, Nanjing University of Information Science and Technology, in 2021. Her research interest is searchable encryption.

Zhihua Xia (Member, IEEE) received the PhD degree in computer science and technology from Hunan University, China, in 2011. He worked successively as a lecturer, an associate professor, and a professor with the College of Computer, Nanjing University of Information Science and Technology. He is currently a professor with the College of Cyber Security, Jinan University, China. He worked as a visiting scholar with the New Jersey Institute of Technology, USA, in 2015, and a visiting professor with Sungkyunkwan University, Korea, in 2016. He serves as a managing editor for IJAACS right now. His research interests include AI security, secure computation, digital forensic, etc.

Zhetao Li (Member, IEEE) received the BEng degree in electrical information engineering from Xiangtan University, in 2002, the MEng degree in pattern recognition and intelligent system from Beihang University, in 2005, and the PhD degree in computer application technology from Hunan University, in 2010. He is a professor with the College of Information Science and Technology, Jinan University. He is a member of CCF.



Jian Weng received the PhD degree in computer science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2008. He is currently a professor and the dean with the College of Information Science and Technology, Jinan University, Guangzhou, China. His research interests include public key cryptography, cloud security, and block-chain. He was the PC co-chairs or PC member for more than 30 international conferences. He also serves as an associate editor for the *IEEE Transactions on Vehicular Technology*.