Journal on Big Data
DOI:10.32604/jbd.2021.015892
Article

Tech Science Press

OPPR: An Outsourcing Privacy-Preserving JPEG Image Retrieval Scheme with Local Histograms in Cloud Environment

Jian Tang, Zhihua Xia[*], Lan Wang, Chengsheng Yuan and Xueli Zhao

Nanjing University of Information Science & Technology, Nanjing, 210044, China
[*]Corresponding Author: Zhihua Xia. Email: xia_zhihua@163.com

Received: 10 September 2020; Accepted: 20 December 2020

Abstract: As the wide application of imaging technology, the number of big image data which may containing private information is growing fast. Due to insufficient computing power and storage space for local server device, many people hand over these images to cloud servers for management. But actually, it is unsafe to store the images to the cloud, so encryption becomes a necessary step before uploading to reduce the risk of privacy leakage. However, it is not conducive to the efficient application of image, especially in the Content-Based Image Retrieval (CBIR) scheme. This paper proposes an outsourcing privacy-preserving JPEG CBIR scheme. We design a set of JPEG format-compatible encryption method, making no file expansion to JPEG files. We firstly combine multiple adjacent 8 × 8 DCT coefficient blocks into big-blocks. Then, random scrambling and stream encryption are used on the binary code of DCT coefficients to protect the JPEG image privacy. The task of extracting features from encrypted images and retrieving similar images are done by the cloud server. The group index histograms of DCT coefficients are extracted from the encrypted big-blocks, then the global vector is produced to represent the JPEG image with the aid of bag-of-words (BOW) model. The security analysis and experimental results show that our proposed scheme has strong security and good retrieval performance.

Keywords: JPEG image retrieval; DCT coefficients; BOW; format-compatible

1 Introduction

Along with the gradual maturity and widely applied of multimedia technology and imaging device, vast number of HD images are produced by enterprises and individuals every day. As local users usually don't have enough memory to store these storage-consuming images, they will choose to outsource images to the cloud server. The revolution of cloud computing in IT industry has focused local users' attention on cloud products, such as AWS, Microsoft Azure, Google Cloud Service, Baidu Net disk and Ding Net disk, which can offer rich computing and storage resources.

While people are benefiting of cloud service, they are facing the hazard of privacy leak or network violence. Outsourcing privacy-rich images to the cloud means that the picture may be attacked. Recently, a technology news website reported that thousands of private videos on software zoom have been exposed to public websites [1]. Bloomberg published that Google cloud server leaked 120 million people's information, which alarm people on maintaining information security [2]. Various events indicate that the security of the cloud server needs to be improved. The most common measure can be adopted is encrypting the images before uploading them to the cloud. However, encrypting the images will affect the performance of image processing, including image retrieval. In addition, it is not easy to extract effective features while keeping good safety performance.

In general, image searchable encryption methods consist of two types. The first one is the feature-encryption based scheme and the second one is the image-encryption based scheme. For the first one, image features are extracted before image encryption, which increases the computing cost of local users. For image-encryption based schemes, when the image owners want to retrieve the similar image, they only need to encrypt the images and upload them to the cloud. Feature extraction, index building and image search are done by the cloud server, reducing the use's workload.

Contribution: We design an outsourcing secure JPEG image retrieval scheme in this paper. This format-compatible scheme can achieve perfect retrieval performance and get strong security without causing file size expansion. This scheme has contributions as shown below:

1) A specifically-designed image encryption method is presented, including four steps, i.e., VLI binary code encryption, quantization tables encryption, inter-big-block permutation and intra-big-block permutation. This encryption method can well protect the image content, support the efficient extraction of image feature, keep format compatible and cause no file size expansion. In this way, image owners only need to encrypt the image and upload it, while other work such as feature extraction and search are left to the cloud, which greatly facilitates the image owners.
2) The local histograms of the group index of DCT coefficients can be directly calculated as local feature vectors by the cloud server, without any communication to the image owner. With these local feature vectors, the cloud server can generate a global feature vector for each image using the bag-of-words (BOW) model. Then, the similarity between the images can be measured by calculating the Manhattan distance between the global feature vectors. It is shown that our scheme can reach better retrieval performance than the state-of-art ones not only in JPEG-domain, but also in spatial domain according to the experimental results.

2 Related Works

Content-Based Image Retrieval (CBIR) liberates people from the tedious work of using text annotation to retrieve images. CBIR schemes can search similar images by calculating the similarities between the image features and get high retrieval accuracy [3–4]. However, CBIR schemes in plaintext domain cannot avoid privacy disclosure. So recently, researchers proposed many CBIR technologies in ciphertext domain. The existing privacy-preserving CBIR (PPCBIR) schemes consist of two types: The feature-encryption based schemes and the image-encryption based schemes.

For the first method (the feature-encryption based image retrieval scheme), image owners need to extract the features, and encrypt the images and their features before uploading them to the cloud. Lu et al. [5] had the first attempt at image retrieval in encrypted domain based on CBIR technologies. They used the methods of order preserving encryption and hash functions to avoid the privacy disclosure caused by the analysis of image content by cloud server. A secure indexing framework has also been developed to ensure good search results. In [6], Lu et al. proposed three encryption methods, all of which can avoid the disclosure of image privacy. But the disadvantage is that the image retrieval accuracy is not high enough when these protection methods are used. For improving the retrieval accuracy [7], homomorphic encryption method is proposed to protect the image features. This encryption methods can achieve good retrieval accuracy and security, but increases the communication cost between the image owner and the cloud server. In [8], a large-scale encrypted image retrieval scheme was proposed by Weng. They utilized robust hash values and certain omittance to improve the security. The level of security can vary with different policies. In [9], Xia et al. used earth mover's distance to calculate the similarity between SIFT features. Locally sensitive hashes are used to reduce time complexity. In [10], Xia et al. used MPEG-7 to represent the images. A secure KNN method and watermarking were jointly proposed to keep images' safety. Locality-sensitive hash was utilized to improve the efficiency in the retrieval system. In [11], Yuan et al. also utilized the secure KNN to avoid the sensitive information reveal, and they built a tree index to increase the search efficiency. They have tried to outsource the tree index to the cloud, but leads the more communication burden to the image owner.

The feature extraction of the above schemes is done by the image owner. So as to reduce the workload of image owners, many image-encryption based schemes are proposed recently. The image owner only needs to encrypt the image and upload it, and the task of feature extraction is left to the cloud server. It is very important to design a simple and effective encryption method and can extract features from the encrypted images. Xu et al. [12] presented a privacy-preserving image search scheme using the orthogonal decomposition. AES algorithm is applied to encrypt the AC coefficient, which protect the image content. And the rest part of the information is used for similarity calculation. Bernardo et al. [13] designed a novel image encryption framework to secure the CBIR service. They protect the image privacy with pixel value substitution and encryption. And the global color histograms are used for the encrypted image retrieval. In [14], Xia put forward a novel scheme using three encryption methods together to protect the privacy of image content. Profiting from the impact of BOW model, the image search accuracy was improved with the help of visual words produced by clustering. But the image encryption in the airspace will destroy the correlation between pixels and result in inefficient compression, which is not conducive to the storage and transmission of encrypted images. In [15], Zhang et al. encrypt the image in the JPEG compression process using stream encryption, so it can guarantee the compatibility of the format and do not affect the compression of the encrypted images. Cheng et al. used Markov model and SVM in [16], and utilized the statistics of (r, v) pairs in each 8×8 block in [17] for image retrieval. These schemes [15–17] are all encrypted in JPEG domain, so they can achieve good compression performance for encrypted JPEG images. But these schemes directly extract global Markov features for image searching, so their accuracy results are not very high.

We design a novel secure JPEG image retrieval framework in this paper. Stream encryption and random scrambling methods are applied together to avoid privacy leaks, which can ensure format-compatible and cause no file expansion. The group index histograms of DCT coefficients are extracted from the encrypted big-blocks as local features by the cloud server, along with BOW model to improve the retrieval accuracy.

3 Preliminaries and System Model

3.1 JPEG Compression

The image encryption method proposed in this paper is carried out in the process of JPEG compression. Because of the high compression ratio and good image quality, JPEG becomes popular for image storing and transmitting [18]. Here we briefly introduce the process of JPEG image compression.

1. **Color space transformation.** First of all, we need to transform the color from RGB into YUV, which represent the images through luminance and chrominance component.
2. **Image block division.** Divide the image into 8×8 blocks in Y, U, and V component as independent units for following coding.
3. **Discrete cosine transformation.** Discrete cosine transformation is happened in each 8×8 blocks, generating 64 DCT coefficients which are denoted as d_{ij} where $i, j \in \{0, \cdots, 7\}$.
4. **Coefficient quantization.** Following, the quantized DCT coefficients are generated as $d_{ij} \leftarrow \frac{d_{ij}}{Q_{ij}}$, where Q_{ij} represent the elements in the quantization tables,$i, j \in \{0, \cdots, 7\}$.
5. **Intermediate encoding.** The quantized coefficients are scanned, generating a one-dimensional vector in each 8×8 block. The difference of the quantized DC coefficient in current block and the former block are calculated, denoted as $(-, v)$. The quantized AC coefficients before the last nonzero coefficient are encoded to be run-length and value pairs, denoted as (r, v).
6. **Entropy encoding.** According to Tab. 1, a group index id_g and a VLI binary code C_b can be produced from the value v in (r, v) pair. As a result, the quantized DCT coefficients are encoded to be triplets, denoted as (r, id_g, C_b). The run-length r and the group index id_g can be later encoded into Huffman binary code according to the Huffman tables.

In the end, these binary streams converted from quantized DCT coefficients and other image information compose the JPEG file, as shown in Fig. 1. SOI and EOI are the start and the end marks for JPEG image files. The header section contains information such as the size of the image and so on. In this scheme, we bundle non-overlapping 8 × 8 image blocks into big-blocks, which are represented as $Bblk$.

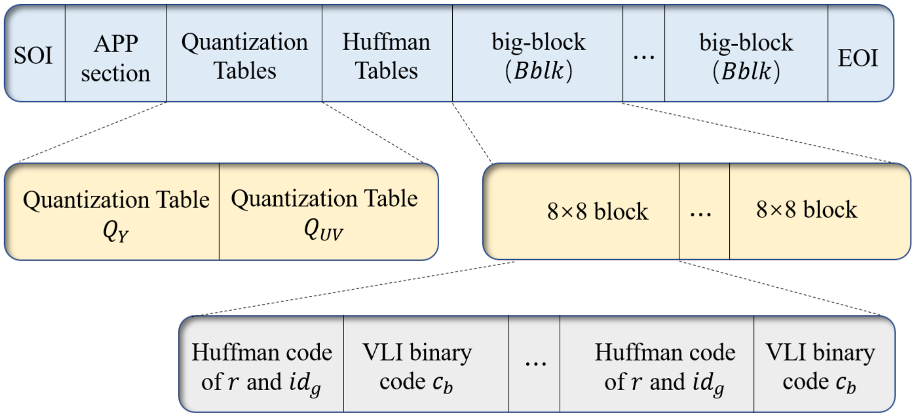


Figure 1: The structure of JPEG image files

Table 1: VLI coding table

Value	id_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
...	...	...
32767，... ，32767	15	...

3.2 Bag-of-Words Model

We use feature vectors to represent the image, and calculate similarities between images through the distance between the feature vectors. The BOW model consists of three steps:

1. **Local feature generation.** There are many kinds of local features for image retrieval in plaintext domain. In our scheme, we extract the histogram of VLI code length (id_g) at different JPEG frequency positions in each big-block $Bblk$ as local feature.
2. **Vocabulary construction.** All the local features extracted from the image are clustered, the clustering centers are defined as visual words. In this scheme, we use k-means for clustering.
3. **Global feature generation.** Finally, every local feature can be represented by its nearest visual word. In this way, each image can be represented by a histogram of visual words, which is a global feature for retrieval. In addition, normalization can eliminate the influence of image size.

3.3 System Model

There are two roles in system model for this paper: one is the image owner and another one is the cloud server, as shown in Fig. 2. Firstly, the image owner encrypts the images before uploading them. After receiving the encrypted images, the cloud server stores it and extracts the features. When the image owner wants to retrieve similar images, the query image is encrypted and uploaded to the cloud server. The feature of the query image is also extracted by the cloud server and matched with the features of the images in the cloud. Eventually, the images similar to the querying image are returned to the image owner for decryption.

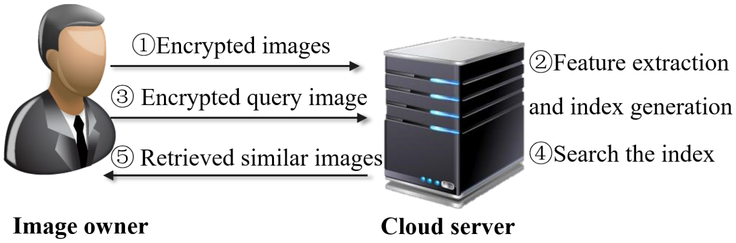


Figure 2: System model

4 The Proposed Scheme

In the proposed scheme, we present a set of image encryption methods to prevent the disclosure of image information. In addition, we design a local features extraction method performed by cloud server. Local features of the encrypted image are then clustered in the BOW model to generate the global features for better retrieval accuracy.

4.1 Image Encryption

In order to prevent the disclosure of image information, we present four steps of encryption methods for the image owner, i.e., VLI binary code encryption, quantization tables encryption, inter-big-block permutation, and intra-big-block permutation. In the preparation, we produce secret keys for the later encryption process.

4.1.1 Secret Key Production

In the initial stage, the image owner has a security key sk. Next, image owner utilizes pseudorandom function(prf) and a pseudorandom permutation generator$(prpg)$ to generate the keys as following,

$$\begin{cases} key_{C_b} \leftarrow prf(sk, ID, C_b) \\ \{key_{Q_*}\}_{*\in\{Y,UV\}} \leftarrow prf\left(sk, ID, \{Q_*\}_{*\in\{Y,UV\}}\right) \\ pmt_B \leftarrow prpg(sk, ID, Bblknum) \\ pmt_b \leftarrow prpg(sk, ID, blknum) \end{cases} \tag{1}$$

where ID is the image's identity, C_b represents the VLI binary code, $Bblknum$ denotes the number of big-blocks, and$blknum$ denotes the number of 8 × 8 blocks.

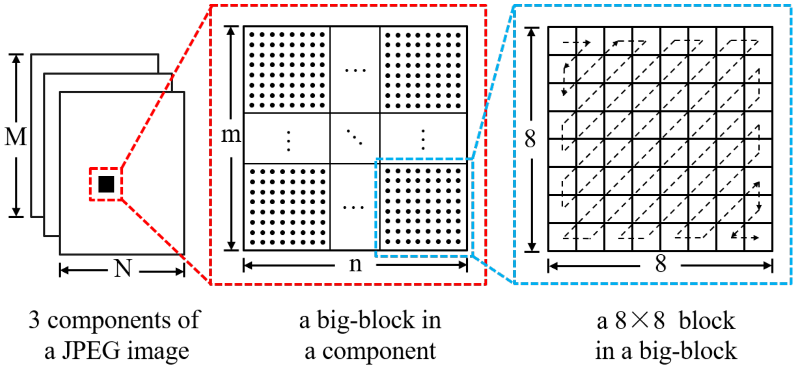

M
N
m
n
8
8
3 components of a JPEG image
a big-block in a component
a 8×8 block in a big-block

Figure 3: The construction of big-block

4.1.2 VLI Binary Code Encryption

As mentioned in 3.1, through decoding JPEG images, the VLI binary code can be respectively extracted in three components and encrypted by xor operation. Formula is as follows:

$$C_b^{'} \leftarrow C_b \oplus key_{C_b} \tag{2}$$

4.1.3 Quantization Tables Encryption

We can know from 3.1 that two quantization tables in JPEG header file can be obtained by decoding. We encrypt the quantization tables to prevent privacy leakage by stream encryption. It is worth noting that the encryption of the quantization tables will not affect the feature extraction. The quantization tables are encrypted as

$$Q_*^{'} \leftarrow Q_* \oplus \{key_{Q_*}\}_{*\in\{Y,UV\}} \tag{3}$$

4.1.4 Big-Block Permutation

In this scheme, for protecting the image privacy and supporting feature extraction, we assemble the adjacent 8 × 8 blocks to be the big-blocks in each component for permutation. As shown in Fig. 3, we denote the i-th big-block in an image as $Bblk_i$. Due to the downsampling in the JPEG compression process, the *U* and *V* components have the half size to the *Y* components in both the height and width. So, the big-blocks in *U* and *V* component are half size to the one in *Y* components in both the height and width. The big-block permutation is as follows:

$$Bblk_i \leftarrow Bblk_{pmt_B[i]} \tag{4}$$

4.1.5 8 × 8 Block Permutation

In order to further protect the image content, we permutate the order of 8×8 blocks within the big-block in each component. Denote the j-th 8 × 8 block in an image as blk_j. The encryption formula of 8 × 8 block permutation is:

$$blk_l^{'} \leftarrow blk_{pmt[j]} \tag{5}$$

where $j = 1, \cdots, blknum$, and $blknum$ is the number of 8 × 8 blocks in a DCT coefficient matrix.

The image owner encrypts all of his images by the above steps. Then, the encrypted image set can be uploaded to the cloud server.

4.2 Image Feature Extraction

In our proposed scheme, feature extraction and retrieval are done by cloud services, which greatly reduces the burden on the image owner. Feature extraction method in this scheme contains four steps here. In the first, we need to preprocess the image data. Local histogram features are then calculated from the big-blocks $Bblk$. Next, Local histogram features are clustered into visual words using *k*-means algorithm. Finally, every image can produce a normalized occurrence histogram of visual words to represent the image.

4.2.1 Data Preprocessing

Even after the previous four steps of encryption by the image owner, the id_g histogram of the encrypted JPEG image at different frequency positions is the same as that of the plaintext image. This invariant information can be used for image search.

By decoding the secret JPEG image, we can get the quantified DCT coefficient in *Y*, *U*, and *V*. Then, according to Tab. 1, we can convert three components into three Group index matrixes. Then, we will truncate the value of three group index matrices according to their probability distribution to reduce redundancy. The three group index matrices after the truncation can be respectively denoted by G_Y, G_U, and G_V.

4.2.2 Local Feature Generation

We extract the group index histograms of different frequency positions in each component to represent local features. The formula to calculate a local feature in a big-block is as follows:

$$F(x) = \frac{\sum \sigma(x=t)}{blknum} \tag{6}$$

where t = 1,...,64. And $\sigma(\Delta)$ = 1 if Δholds, else $\sigma(\Delta)$ = 0.

For each big-block, we concatenate three kinds of features to generate a feature vector with $64 \times 3 \times (1+\tau)$elements, which is defined as the local feature, we set the truncation parameter τ = 8 in this scheme.

4.2.3 Vocabulary Generation

In this way, each encrypted image can be presented by a set of local features. We cluster all of these local features in the whole image database and produce k cluster centers, which is a variable parameter. We defined the k cluster centers as visual words.

4.2.4 Histogram Calculation

Each image can be presented by a set of local features according to the vocabulary, so each local feature can be replaced by the nearest visual word. And then, each image can produce a global feature by calculating the occurrence histogram of the visual words.

4.3 Image Search

Similar images search is outsourced to the cloud server without any communication with image owners. After receiving a query image encrypted with the same four methods as mentioned in Section 4.1, the cloud server will extract the encrypted query image's local feature through the methods mentioned in Section 4.2, and generate a global feature according to the visual words of the image set. Finally, the global feature of the encrypted query feature can be compared with other global features in image dataset, by calculating the Manhattan distance. The formula is as follows:

$$d(Query, Image) = \sum_{t=1}^{k} |Query_t - Image_t| \tag{7}$$

where $Query$ and $Image$ separately represent the global feature of the query image and encrypted images in database, which are both k-dimensional vectors. The smaller the Manhattan distance between the global features, the more similar the corresponding image. In this way, a certain number of images are sent to the image owner for the next decryption using the secret key.

5 Security Analysis

In our scheme, the cloud server completes the tedious work of feature extraction and image retrieval. The honest-but-curious cloud server, which is the only potential adversary, might analyze the contents of the encrypted images to gain privacy. We interpret the safety of the proposed scheme under COA. As we generate a unique key for each image. As each image corresponds to a unique key, the threat can only be a brute-force attack.

Summary of information leakage. The cloud server knows part of the encrypted image, for supporting the feature extraction and encrypted images comparison, such as the image size, the size of big-block $Bblksize$, the number of big-block $Bblknum$, the length of VLI binary code c_B and the size of quantization tables.

Theorem 1. If the scheme is attacked under the COA model, the security intensity Sec is with an honest-but-curious probabilistic polynomial time (PPT) adversary under the COA model, the encrypted image is computational secure and the security strength equals to

$$Sec = len_{Cb} + len_{Q_Y} + len_{Q_{UV}} + log_2^{Bblknum!} + 3 \times Bblknum \times log_2^{Bblksize!} \tag{8}$$

where len_{C_b}represents the total length of VLI binary code, $len_{Q_Y}$and $len_{Q_{UV}}$ respectively represents the length of two quantization tables, $Bblksize$ is the size of big-blocks and $Bblknum$ is the number of big-blocks in an image.

Proof. Because different encrypted images correspond to different keys, the only threat is the brute-force attack. So, the security strength depending on the total length of secret key. Xor operation is first conducted on the VLI binary code C_b, where length is len_{C_b}. Second, xor operation is conducted on two quantization tables, where length is sum of length of Q_Y and Q_{UV}. Third, inter-big-blocks permutation is conducted. The encryption complexity is $log_2^{Bblknum!}$. In the end, intra-big-blocks permutation is conducted. The encryption complexity is $3 \times Bblknum \times log_2^{Bblksize!}$. These four parts add up to safety strength.

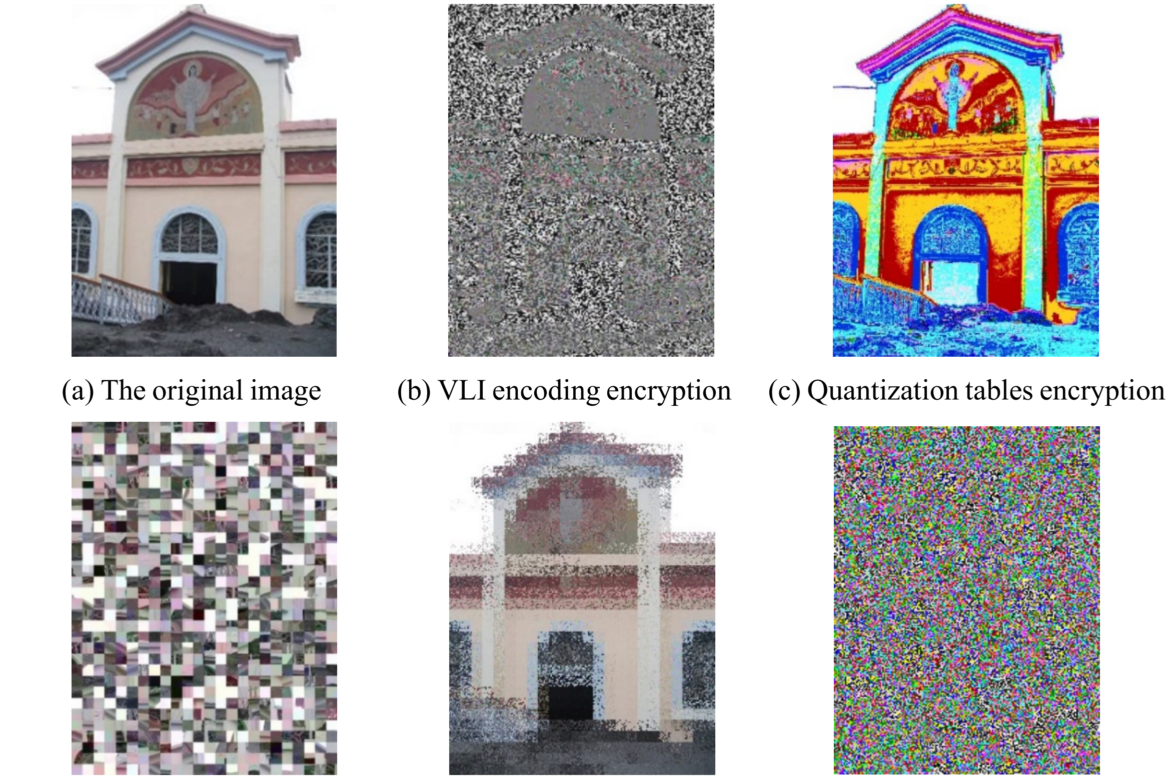
End of Proof.

6 Evaluation of Experimental Results

In order to evaluate the performance of the proposed scheme, we carry out the corresponding experiments and analyze the results in four different aspects, i.e., image encryption effectiveness, time consumptions, retrieval accuracy, and expansion of file size. All tests would be operated in the Inria Holidays dataset [19], consisting of 1491 color images with different sizes and has 500 classes. Inria Holidays database provides a Python evaluation package to calculate mAP and was used in many image retrieval schemes, which facilitates fair retrieval accuracy comparison. We use MATLAB R2018b to implement the model in Ubuntu 18 system with 64 GB of RAM.

6.1 Image Encryption Effectiveness

To protect the privacy of users' images, four encryption steps are applied in the proposed scheme. As shown in Fig. 5, the single or combined visual effects of the four encryption steps are revealed. Fig. 5(b) is the VLI binary code encryption, (d) inter-big-block permutation can protect the image content well, while the (c) quantization tables encryption only disturbs the color information, but the intra-big-block permutation (e) is easy to leak the image privacy. In sum, the image content can be perfectly protected by the combination of these four encryption steps (f).



(a) The original image (b) VLI encoding encryption (c) Quantization tables encryption

(d) Inter big-block permutation (e) Intra big-block permutation (f) Joint encryption

Figure 5: Encrypted images

Table 2: The time consumptions of encryption

Encryption steps	Time consumption (s)
Encryption of VLI binary code	4.8094
Encryption of quantization tables	0.0010
Permutation of inter-big-blocks	0.0759
Permutation of intra-big-blocks	0.3008
The combination of four steps	5.1871

Table 3: The time consumptions of local feature extraction (s)

$Bblksize$	32 × 32	64 × 64	96 × 96	128 × 128	160 × 160
Y	22.094	8.675	6.836	8.579	34.886
U	22.137	6.217	4.748	4.865	31.542
V	21.947	6.093	4.925	4.927	30.773

Table 4: The time consumptions of K-means (s)

	k	32 × 32	64 × 64	96 × 96	128 × 128	160 × 160
Y	500	283.968	1270.568	376.893	905.940	97.049
	1000	409.342	1691.957	592.028	267.125	159.467
	3000	668.692	2623.956	568.416	300.933	163.774
	5000	562.965	1383.001	729.191	412.373	266.813
U	500	169.861	515.580	408.944	303.892	103.373
	1000	301.670	1292.607	616.361	273.854	167.228
	3000	696.147	3108.926	636.486	280.136	167.654
	5000	611.706	2038.526	1023.308	411.658	287.318
V	500	152.616	486.598	394.292	179.433	149.588
	1000	288.639	1077.425	642.591	286.932	146.613
	3000	713.361	2799.733	636.535	276.343	173.269
	5000	1409.675	1986.386	874.248	399.774	279.038

6.2 Time Consumptions

The time consumption for each step of image encryption (as shown in Tab. 2), local features generation (as shown in Tab. 3), K-means clustering (as shown in Tab. 4), global features production (as shown in Tab. 5), and search (as shown in Tab. 6) are recorded here.

It can be seen that the time spent on each step of the experiment is not too much. The step of local features generating and clustering can be performed automatically by the server in the cloud which means it does not affect the user.

6.3 Retrieval Accuracy

We utilize mAP (mean average precision) to reflect the accuracy of image retrieval, with the aid of a python evaluation package in Inria Holidays image set. We individually and collectively test the mAP values in Y, U, and V component. We also select different parameters such as the number of cluster centers (k) and the size of the big-block ($Bblksize$). As shown in Tab. 7, when k is set to be 3000 and $Bblksize$ is set to be 64 or 96, it reaches the highest mAP. In addition, we also compared our mAP values with those of other schemes as shown in Tab.8.

Table 5: The time consumptions of global feature generation (s)

	k	32 × 32	64 × 64	96 × 96	128 × 128	160 × 160
Y	500	441.757	138.577	71.685	110.413	42.872
	1000	792.698	290.989	140.602	91.207	73.834
	3000	2099.168	425.185	195.690	122.119	81.258
	5000	22131.733	730.1736	331.912	179.871	124.612
U	500	435.922	118.304	78.037	51.123	43.796
	1000	783.504	237.644	140.023	90.864	71.607
	3000	2092.387	443.753	186.861	113.360	82.206
	5000	2461.563	659.535	397.587	176.415	124.612
V	500	434.035	118.304	78.037	51.123	87.954
	1000	782.760	233.764	150.425	90.907	65.662
	3000	2088.867	414.067	194.943	113.866	82.544
	5000	1409.634	609.631	292.802	178.749	122.003

Table 6: The time consumptions of search (s)

	k	32 × 32	64 × 64	96 × 96	128 × 128	160 × 160
Y	500	4.008	3.934	4.884	10.038	5.009
	1000	4.567	7.202	7.150	7.275	6.892
	3000	13.198	18.450	14.097	20.757	14.097
	5000	12.774	12.765	18.738	14,955	13.181
U	500	3.475	4.467	4.826	4.910	5.133
	1000	6.249	7.154	7.251	7.212	7.298
	3000	15.189	23.021	8.157	8.517	8.1577
	5000	34.207	12.434	27.877	12.296	8.157
V	500	4.775	4.406	3.509	4.844	9.967
	1000	5.316	5.976	7.479	7.224	6.250
	3000	14.948	18.977	8.199	14.381	8.199
	5000	34.851	12.248	12.281	12.323	17.063
YUV	500	4.983	4.842	4.901	4.908	4.883
	1000	8.266	6.217	8.183	8.165	8.191
	3000	36.850	36.827	36.904	36.625	37.033
	5000	70.924	60.369	60.468	60.193	59.834

6.4 Expansion of File Size

All four encryption steps of this scheme are operated on the JPEG bitstream, including VLI binary code encryption, inter- big-block permutation, quantization tables encryption, and the intra-big-block permutation. So, the size of the memory occupied by the image will not be changed after the encryption, which is proved by the following data (as shown in Tab. 9).

Table 7: MAP values

	k	32 × 32	64 × 64	96 × 96	128 × 128	160 × 160
Y	500	0.42581	0.43503	0.42469	0.40555	0.39859
	1000	0.44152	0.45215	0.43651	0.43188	0.43209
	3000	0.44681	**0.46744**	0.45774	0.44675	0.44414
	5000	0.45194	0.46017	0.46724	0.45605	0.44161
U	500	0.46302	0.48358	0.46468	0.46072	0.45373
	1000	0.47449	0.49338	0.49197	0.48328	0.45433
	3000	0.48183	0.50736	**0.50804**	0.48299	0.48949
	5000	0.47841	0.50435	0.50510	0.49905	0.49377
V	500	0.44333	0.46806	0.47347	0.45331	0.45629
	1000	0.45945	0.50694	0.47569	0.46924	0.46220
	3000	0.47005	**0.50339**	0.45774	0.48986	0.48771
	5000	0.46937	0.50450	0.50328	0.49066	0.48378
YUV	500	0.54068	0.54074	0.54199	0.53648	0.53605
	1000	0.55517	0.55197	0.55534	0.55137	0.55840
	3000	0.54538	**0.56517**	0.56094	0.55341	0.55010
	5000	0.54363	0.55847	0.55952	0.55676	0.55007

Table 8: The comparison of mAPs with previous schemes

Schemes	mAP
Our Scheme	**0.56517**
Partial-encryption based scheme [12]	0.56040
IES [13]	0.54564
Cheng et al. [16]	0.54187
Cheng et al. [17]	0.36000

Table 9: The comparison of file size with previous schemes

Schemes	File size (GB)
Our Scheme	2.65
IES [13]	20.02
BOEW-YUV [14]	17.97

7 Conclusions

In this paper, we propose a novel encrypted JPEG image retrieval framework, which can Privacy can be guarantee the security, efficiency and accuracy. Stream encryption and scrambling encryption are used together to encrypt the images, ensuring the compatibility of the format and making no change of the image. Each big-block can be represented as a local color histogram by the cloud server for similarity measure. We also use BOW model to cluster these local features to generate a global feature for each encrypted image to obtain good retrieval accuracy. In this proposed framework, the cloud server provides the following services: Image storage, feature generation, and image retrieval, reducing the operations of the image owner. In future work, it is a significant research to use stronger encryption method to protect the security while ensuring the retrieval accuracy.

Funding Statement: This work is supported in part by the Jiangsu Basic Research Programs–Natural Science Foundation under Grant No. BK20181407, in part by the National Natural Science Foundation of China under Grant Nos. 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 Nos. U1836208, 61702276, 61772283, 61602253, and 61601236, in part by National Key R&D Program of China under Grant No. 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.

References

[1] K. Mehrotra, "Zoom faces a privacy and security backlash as it surges in popularity," [Online]. Available: https://www.theverge.com/2020/4/1/21202584/zoom-security-privacy-issues-video-conferencing-software-coronavirus-demand-response.

[2] M. Kartikay, "A billion people's data left unprotected on google cloud server," [Online]. Available: https://telecomlive.com/web/a-billion-peoples-data-left-unprote cted-on-google-cloud-server/.

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

[4] L. Dong, Y. Liang, G. Kong, Q. Zhang, X. Cao *et al.,* "Holons visual representation for image retrieval," *IEEE Transactions on Multimedia,* vol. 18, no. 4, pp. 714–725, 2016.

[5] W. J. Lu, A. Swaminathan, A. L. Varna and M. Wu, "Enabling search over encrypted multimedia databases," *Media Forensics and Security International Society for Optics and Photonics*, 2009.

[6] W. J. Lu, A. L. Varna, A. Swaminathan and M. Wu, "Secure image retrieval through feature protection," in *Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing*, pp. 1533–1536, 2009.

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

[8] L. Weng, L. Amsaleg, A. Morton and S. Marchandmaillet, "A privacy- preserving framework for large-scale content-based information retrieval," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 152–167, 2014.

[9] Z. H. Xia, Y. Zhu, X. M. Sun, Z. Qin and K. Ren, "Towards privacy-preserving content-based image retrieval in cloud computing," *IEEE Transactions on Cloud Computing*, pp. 1, 2015.

[10] Z. H. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun *et al.,* "A privacy- preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.

[11] J. W. Yuan, S. C. Yu and L. K. Guo, "SEISA: Secure and efficient encrypted im- age search with access control," in *2015 IEEE Conf. on Computer Communications (INFOCOM)*, pp. 2083–2091, 2015.

[12] Y. Y. Xu, J. Y. Gong, L. Z. Xiong, Z. Q. Xu, J. W. Wang *et al.,* "A privacy-preserving content-based image retrieval method in cloud environment," *Journal of Visual Communication and Image Representation*, vol. 43, pp. 164–172, 2017.

[13] B. Ferreira, J. Rodrigues, J. Leitao and H. Domingos, "Practical privacy-preserving content-based retrieval in cloud image repositories," *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 784–798, 2019.

[14] Z. H. Xia, L. Q. Jiang, D. D. Liu, L. H. Lu and B. Jeon, "Boew: A content-based image retrieval scheme using bag-of-encrypted-words in cloud computing," *IEEE Transactions on Services Computing*, 2019.

[15] X. P. Zhang and H. Cheng, "Histogram-based retrieval for encrypted jpeg images," in *2014 IEEE China Summit & Int. Conf. on Signal and Information Processing (ChinaSIP)*, pp. 446–449, 2014.

[16] H. Cheng, X. P. Zhang, J. Yu and F. Li, "Markov process-based retrieval for encrypted JPEG images," *EURASIP Journal on Information Security*, vol. 2016, no. 1, pp. 417–421, 2016.

[17] H. Cheng, X. P. Zhang, J. Yu and Y. Zhang, “Encrypted jpeg image retrieval using block-wise feature comparison,” *Journal of Visual Communication & Image Representation*, vol. 40, pp. 111–117, 2016.

[18] G. K. Wallace, “The JPEG still picture compression standard,” *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[19] H. Jegou, M. Douze and C. Schmid, “Hamming embedding and weak geometric consistency for large scale image search,” in *ECCV'08 Proc. of the 10th European Conf. on Computer Vision: Part I*, pp. 304–317, 2008.