Received: 10 April 2022 | *Revised:* 21 May 2022 | *Accepted:* 21 May 2022
DOI: 10.37917/ijeee.18.2.8 *Vol. 18| Issue 2| December 2022*

Open Access

Iraqi Journal for Electrical and Electronic Engineering
Original Article

Privacy-Preserve Content-Based Image Retrieval Using Aggregated Local Features

Ali Lazim Lafta*, Ayad I. Abdulsada
Department of Computer science, Education College for Pure Sciences, University of Basrah, Basrah, 61004, Iraq

Correspondence
* Ayad I. Abdulsada
Department of Computer science
Education College for Pure Sciences,
University of Basrah, Basrah, Iraq
Email: ayad.abdulsada@uobasrah.edu.iq
alialmalki000122@gmail.com

Abstract
Due to the recent improvements in imaging and computing technologies, a massive quantity of image data is generated every day. For searching image collection, several content-based image retrieval (CBIR) methods have been introduced. However, these methods need more computing and storage resources. Cloud servers can fill this gap by providing huge computational power at a cheap price. However, cloud servers are not fully trusted, thus image owners have legal concerns about the privacy of their private data. In this paper, we proposed and implemented a privacy-preserving CBIR (PP-CBIR) scheme that allows searching and retrieving image databases in a cipher text format. Specifically, we extract aggregated feature vectors to represent the corresponding image collection and employ the asymmetric scalar-product-preserving encryption scheme (ASPE) method to protect these vectors while allowing for similarity computation between these encrypted vectors. To enhance search time, all encrypted features are clustered by the k-means algorithm recursively to construct a tree index. Results show that PP-CBIR has faster indexing and retrieving with good retrieval precision and scalability than previous schemes.

KEYWORDS: Privacy-Preserve, Content-Based Image Retrieval (CBIR), Clustering, VLAD, ASPE, Indexing, Secure Search.

I. INTRODUCTION

Content-based image retrieval (CBIR) is a useful method used for many years in several real-world applications like face recognition, object detection, medical detection, to search image collection and find similar images. However, the widespread of digital cameras and smartphones led to generate tremendous image collections. Therefore, the application of traditional CBIR methods will be prohibited since more storage and computing resources are required. Cloud computing can help by providing on-demand access to sufficient storage and computing capabilities for data owners. In, this setting, images will be outsourced to the cloud server, and will not be longer under the supervisor of their owner. The authorized users can use CBIR to contact the cloud server and retrieve similar images. Since images are often tending to be personal and contain sensitive information, the direct outsourcing of them to the cloud represents a big problem for privacy. For example, in the medical application of CBIR, patients' images are not allowed to be disclosed other than a specific doctor. To reduce the risk of compromising privacy, most of the time, images are encrypted before being uploaded to the cloud servers. The direct application of simple encryption schemes will disable CBIR operations. Thus, it is of utmost importance to develop privacy preserver CBIR systems (PP-CBIR) that can deal with encrypted images without decryption.

The current privacy-preserving CBIR schemes work as follows: the data owner extracts some feature vectors from the image. Then, all images and vectors are encrypted before being outsourced to the cloud server. In this setting, the similarity of two images can be calculated by computing the distance between their corresponding encrypted features.

Image feature vectors can be global that will make summery vector to the whole image, or local that will represent the image by its interest points, this will lead to make the image represented by many feature vectors. However, global feature will depend on signal representation to the image, any change in the light, scaling, rotation or color depth in the same image will lead to different feature vector. Many methods used for global feature for example shape [1, 2], color histograms [3] and texture [4, 5], etc. can be used to represent the image .In another hand the local feature will depend on the interest point located in the image like edges, angle or small image patch, etc. Interest points can be more immune to rotations, scaling, color depths and other effect. The interest point will depend not on one single pixel but it

will make use of its neighbors pixels. SIFT [6], SURF [7], ORB [8], and LBP [9], most known local feature image descriptors each image descriptor will have its own length , SIFT have $d = 128$ dimensional with positive values. An Example of local features is illustrated in Fig.1.

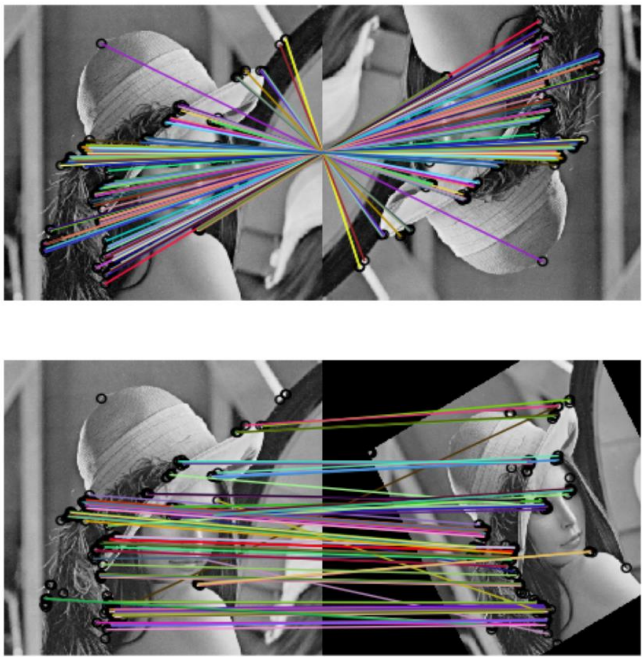


Fig. 1: Local feature: interest points and descriptors

Local Image descriptors representations are used in many applications such as image search and identifications. However, this kind of representations will lead to many feature vectors for signal image. Thus, the image similarity will consume big computational power. Thats way many methods are proposed to manage the big quantity of features vectors.

In this paper, we employed the local feature descriptors aggregations method VLAD[10]. In addition we select two local descriptors (ORB) [8] and (SIFT) [6] as our mean image descriptors .

To protect the aggregated vectors, many PP-CBIR schemes use homomorphic encryption HE, which allows performing some arithmetic operations over the encrypted data. However, HE incurs huge complexity. Instead, we utilized ASPE method [11], introduced by Wong. et al. in 2009. This scheme can implement kNN similarity efficiently in the encryption domain. However, the cloud server has to perform a huge number of operations to compare the provided query against the current encrypted vectors. To mitigate this issue, we implement a hierarchal-indexing method that employs a k-means clustering algorithm, to improve the search efficiency. The data owner has to share the encryption key with authorized data users who generate the trapdoor for their query image. Under this setting, the privacy of the data user is preserved, since the data owner does not know what the user is searching for.

Our Contributions. We will summarize what we contribute in the following points. 1) we aggregate the local feature descriptors into a single aggregated feature vector using the VLAD method. Such treatment will ensure fast searching and indexing. 2) we employ a light encryption scheme, ASPE to protect the aggregated vectors, which gives us very good scalability and efficiency. 3) to enhance search efficiency, we apply k-means algorithm for clustering to generate a hierarchal index. 4) we implement our scheme with two popular local descriptors: ORB and SIFT.

Paper Organization. The remaining are segmented into the following: Section **II** presentation most pertinent PP-CBIR schemes. Section **III** mentioned a brief description to the problem of this work. The section IV will present our proposed scheme. **V** section will display the result and experiment. The paper will be concluded in the **VI** section.

II. Related Works

The current schemes of privacy-preserving CBIR work in one of the following modes: encrypted features schemes, and encrypted image schemes. The first mode extracts image features and encrypts it by the data owner then the data owner will store it in the cloud serves provider, while the second mode encrypts images by the data owner and delegates the feature extraction in the encryption domain to the cloud server-side.

A. *Encrypted features schemes*

In the past years, much research tries to fix the problems of PP-CBIR Abduljabbar, Ibrahim, Hussain, Hussien, Al Sibahee and Lu [12] present EEIRI. In this scheme, the images are represented using local features. We develop and validate a secure scheme for measuring the Euclidean distance between two descriptor sets.. Lu, Swaminathan, Varna and Wu [13] suggest PP-CBIR in the encrypted domain where images are described as global histograms of visual words. Such histograms are encrypted by either order-preserving encryption or min-hash functions in order to find the similarity between two images, Jaccard distance was used to measure the distance between their histograms. Lu, Varna, Swaminathan and Wu [14] extract global features such as color histograms and suggest three methods to encrypt such features: bit randomization, random projection, and random unary encoding. However, all these three methods have shown low retrieval accuracy. Some researchers[15] [16] have utilized homomorphic encryption to get high accuracy. However, the distance calculation between the encrypted query vector and the encrypted outsourced vectors, in the case of HE, requires active communication between the cloud server and the image owner. Abdulsada, Ali, Abduljabbar and Hashim [17] propose an efficient scheme that provides content based search over encrypted image database. Qin et al [18] proposed a secure image retrieval scheme based on the corner descriptor Harris and uses the method of locality-sensitive hash (LSH), which shows a modest retrieval accuracy. Xia, Wang, Zhang, Qin, Sun and Ren [19] proposed to use ASPE to secure global features. To encrypt the features, they used a binary vector to split each feature vector into two vectors. Then, they defined two invertible matrices to encrypt each split feature vector. This will enable the cloud server to calculate the similarity distance between

two image vectors in the encryption domain without any communication between the data owner and the cloud server. However, the authors used global features to describe images, whereas our scheme uses aggregated local features. Xia, Zhu, Sun, Qin and Ren [20] designed a PP-CBIR that will use the Bag of visual words (BOVW) model to represent the image based on SIFT local features. The Earth Mover's Distance (EMD) is used to measure the distance between two images. EMD is calculated by constructing and solving a linear programming problem, The above-mentioned method was used to ensure that sensitive information to be protected does not leak out. However, one of its problems is that it needs to communicate more than once between the data owner and the cloud provider to find images that can be close to the query image, this situation will greatly increase the time taken to complete the search process and incurs high communication cost.

B. *Encrypted image schemes*

Cheng, Zhang, Yu and Zhang [21] proposed a PP-CBIR scheme that works only with JPEG images. In this work, the image is parsed as a JPEG file bitstream of pairs (r, v) (where r is the coefficient values and v is the run-length). To protect the content of an image, a stream cipher is used to hide the quantization tables and the DC coefficients. Furthermore, a permutation is used on the 8*8 DCT coefficient blocks, also used on the (r, v) pairs within the blocks. From the encrypted images, the authors have extracted five descriptive features. The retrieval accuracy of [21] is further improved in [22], where the DC values are just scrambled and the permutation of the DC blocks is discarded. Then, Markov was used to extract features from images. The visual information of images could disturb by the permutation. However, this scheme reveals (r, v) values, which can manipulate the contents in the images. Ferreira, Rodrigues, Leitao and Domingos [23],[24] It uses full image cipher as every pixel in the image is encoded using substitution, and the location of a pixel is relocated using shuffled. From the encrypted image, a global histogram was generated. The hamming distance is used to measure the distance between two images. Wang, Xia, Fei and Xiao [25] proposed to extract random features from images that are encrypted by AES. However, the retrieval accuracy for this combination did not give acceptable accuracy upon examination. Xia, Lu, Qin, Shim, Chen and Jeon [26] had presented a scheme that encrypts images in YUV color space. The DC coefficients of the Y are encrypted by stream cipher, while the AC coefficients and remaining color components are protected by shuffling. From this encrypted image, two histograms are extracted and then concatenated. The first histogram is generated from AC coefficients of Y, while the second histogram is generated from U, V components. Here, the Manhattan method is used to measure the distance between two vectors of two images. Despite this, the accuracy of the search in this scheme was unacceptable.

Xia et al [27] use a method to encrypt the entire image using the substitution method and then use permutation on the encrypted image, Then the cloud provider extracts the features of the encrypted image and represent it in encrypted histograms by using the BOVW model. Such a scheme shows high accurate results but it has a security problem since it uses weak encryption primitives that suffer from statistical attacks. Xia et al. [28] present a secure Local Binary Pattern (LBP) feature extraction system, the pixel and block shuffling are integrated to create a privacy-protected LBP extraction method in the encrypted field. The system gives good efficiency, but the security was compromised. Xu et al [29] proposed PP-CBIR the image is divided into two symmetric parts, the first part is protected using the AES encryption method. As for the second part, it remains the same, It is will be used to extract the image features, according to the above, this method will leak a lot of information about the content of the image.

III. PROBLEM DESCRIPTION

A. *System model*

We divide our proposed scheme into three main entities: the first is the $data\ owner$, the second is the authorized $data\ user$, and the third is the $cloud\ service$ provider as Fig.2.

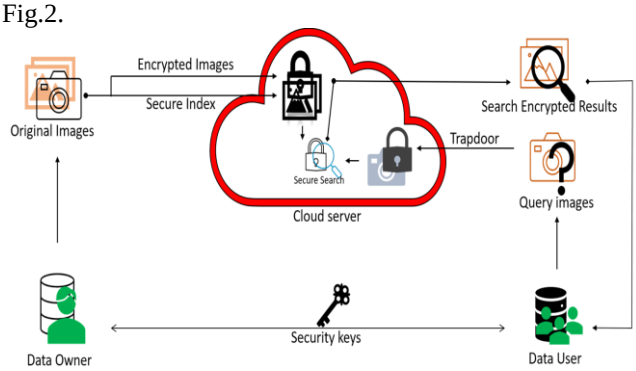


Fig. 2: Proposed scheme PP-CBIR

The Data owner plans to outsource his private image collection $M = \{m_1, m_2, \dots, m_n\}$ of n images in its encrypted format $C = (c_1, c, \dots, c_n)$ to an external cloud server, with the aim of enabling the search over the encrypted collection. In the beginning, the data owner extracts aggregated local feature vectors $V = (v_1, v_2, \dots, v_n)$ from the plaintext image collection, and then create our secure index tree I from V. Then both C and I will be both stored in the cloud server. The data owner should authorize the data users via a specific authentication scheme, which is outside the scope of our work as many existing PP-CBIR schemes [15, 19, 28-31] . Only the authorized users can submit valid search requests to the cloud server. This paper, we consider the setting of a single data owner. When we have multiple owners with different sets of data users, then indexes, image collections, search requests are all encrypted with different keys.

The Data users are the users who authorized by data owner and want to search query images in the encrypted collection. To retrieve images, data user must provide a valid search trapdoor TR to the cloud server. When he/she gets the encrypted results, he/she will use the secret keys provided by the data owner to decrypted the encrypted results .

The Cloud server provides the responsibility to store the encrypted image collection with its encrypted index and

supports computational power needed to answer the search requests of data users.

B. Design Goals

1. *Efficiency*. The search linearly is completely ineffective and impractical by default for huge image collections. Our proposed utilizes a secure tree index to achieve better search efficiency.

2. *Data privacy* the actual content of the image collection, image features, and search requests should remain secret to the semi-trusted cloud server.

C. Thread model

We follow the previous PP-CBIR schemes [24, 30, 32, 33] will deal with the cloud server as *semi-honest* entity, which means that it properly obeys the and implement the protocol but try to obtain extra private information from the communication data. This is why we have to protect carefully the image collection, secure index and search trapdoors.

In this work, we considered that the data users and cloud server will not collude. This assumption will enable to establishment of an efficient scheme. Specifically, the data users will not open to the cloud server any secretly shared information used in the generation of search trapdoors and image decryption. In addition, similar to the previous PP-CBIR schemes [23, 27, 30, 31, 33], our scheme leaks to the cloud server both the *access pattern* with the *search pattern*, where the first one reveals the identities of returned images, while the second one, reveals whether the same image query has been searched or not before.

D. Vector of locally aggregated descriptors

The local features need to be optimized to deal with large image collections, several quantize methods (aggregate) has developed to compact image feature into single descriptors vector in tradeoff with precision. Bag-of-features (BOF) [34] and vector of locally aggregated descriptors (VLAD) [10] most known methods .

BOF is constructed from local descriptors to describe the image in one single vector. So, it will significantly lower the time of search since massive descriptors with many vectors doesn't used in the search. BOF works by creating a vocabulary V contain k centroids $\theta = (\theta_1, \theta_2, \dots, c\theta_k)$ belonging to the feature space. Usually, *k*-means algorithm for clustering is used to generating the vocabulary. Image feature descriptors $f_i \in R^d$ are convert to integer tab between 1 and k depending on the similarity outcome. At that time, a vector with the size of k is generated as histogram to represent each image; here the process of resemblance is abbreviated using an abbreviated form of the image the histograms of each image are used for matching rather than the features of the entire image.

VLAD is much better than BOF, which quantizes the local descriptor to it is nearest visual world without caring for the quantization error. VLAD, on the other hand, will record all the differences between the descriptors and limit the noise that affects the result. Local feature descriptor f is assigned to its nearest centroid as $\theta_i = NN(f)$. The image will be represented by VLAD v of l- dimensions, where $l = k * d$.

$$v_{i,j} = \sum_{f|NN(f)=c_i} f_j - \theta_{i,j.} \quad (1)$$

Where $i = 1, \dots, k$, $j = 1, \dots, d$. Finally, L_2 normalization is applied to VLAD vector. Fig. 3 illustrates the VLAD vector for similar images.

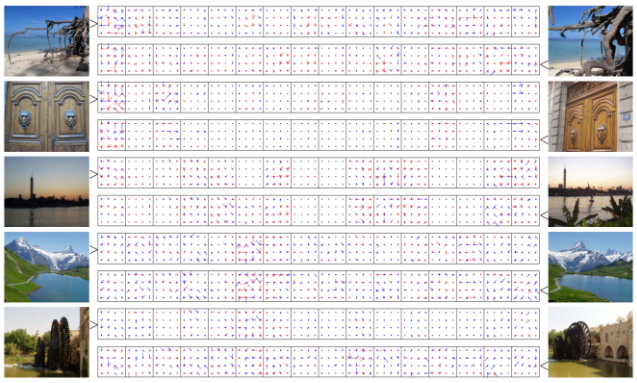


Fig. 3: VLAD descriptors, for $k = 16$ centroids, $d = 128$.

IV. Proposed Scheme

A. Overview of the proposed scheme

The proposed scheme has five main algorithms that are summarized as follows: $KeyGen$, $IndexGen$, $TradoorGen$, $ImgDec$ and $Search$. Given the security parameter λ the data owner runs $KeyGen$ to obtain the secret key set K. He runs the $IndexGen$ algorithm over the image collection M and K to get the encrypted images C and the secure index I. The values of C and I are stored in the cloud server and the set K is shared with the authorized users. When an authorized user wants to retrieve. similar images to his query image q, he/she runs $TradoorGen$ over q and K to generate the secure trapdoor TR, which is submitted to the cloud server. Later on runs $Search$ over TR and I and returns the top-ϕ most similar encrypted images R. Upon receiving R, the data user employs $ImgDec$ to get the plaintext images. Below, we will present our scheme for privacy-preserving content-based image retrieval in more detail.

B. Privacy-preserving CBIR scheme

Please note that the data owner runs $KeyGen$ and$IndexGen$, the data user runs $TradoorGen$ and $ImgDec$, and the cloud server runs $Search$. In this subsection, we explain these algorithms in detail.

- $K \leftarrow KeyGen(\lambda)$ algorithm will receives the security parameter λ and returns the set key K=(S, M_1, M_2, $kcoll$), where is a binary vector of $(l + 1)$ bits. M_1 is an invertible matrix of size $(l + 1)\times (l + 1)$. M_2 is defined in the same of M_1. $kcoll$ it is the secret key that will be used for encryption and decryption of images and image features.
- $(C, I) \leftarrow IndexGen(K, M)$ this algorithm takes as inputs K and M and returners the encrypted image collection C and the secure index I. Images could be

encrypted using any secure method like AES. Thus its encryption is not discussed in this work. The process of index generation will be achieved via two steps.

1) Index generation

For each image $m_i \in M$ the data owner extracts the set of local feature vectors (descriptors) $F = (f_1, f_2, \ldots, f_z)$, where each $f_i \in R^d$. The feature set F is aggregated by using VLAD (which is described in subsection III-D) into a single vector v of size l. In this setting, each image is described as a single vector, which will be used for the search purpose. However, this one-to-one index will be impractical when we have a large number of images. In order to speed up the search, we created a tree-index. Such an index is based on the *k*-means clustering algorithm, where *k* is a user-defined variable, that is the number of centroids that describe the whole image vectors. To reduce searching time, we recursively apply a k-means clustering algorithm to build a tree index. In this context, images (aggregated descriptors) within the same cluster are probably to be similar and we can discard the dissimilar images that are distant from the query image during the search quickly. During the search, the query image will be compared only with the nearest centers recursively. Finally, the cloud server will compute the similarity scores for all images within the remaining cluster. This treatment drastically improves the computation cost.

2) Index encryption

The image aggregated descriptors may leak some information about its actual content. On this basis, the data owner must encrypt the aggregated descriptors before it is sent to the cloud service provider. The encryption method should allow for the encrypted descriptors to be ranked and retrieved without decryption. Homomorphic encryption[35] is usually used to achieve this goal but at the expense of more searching time and additional communication rounds between data user and cloud server. Alternatively, ASPE algorithm [11] is used to protect the aggregated descriptors. To do so, we first extend the aggregated image feature vector $\mathrm{v_i} = (\mathrm{v_{i,1}}, \ldots., \mathrm{v_{i,l}})^{\mathrm{T}}$ into $\hat{\mathrm{v}}_\mathrm{i} = \left(\mathrm{v_{i,1}}, \mathrm{v_{i,2}}, \ldots, \mathrm{v_{i,l}}, \left|\left|\mathrm{v}\right|\right|^2\right)^{\mathrm{T}}$ where $\left|\left|\mathrm{v_{i,l}}\right|\right|$ is the Euclidean norm of $\mathrm{v_i}$, then we will divide the $\hat{\mathrm{v}}_\mathrm{i}$ into two random vectors ($\hat{\mathrm{v}}_{\mathrm{ia}}$, $\hat{\mathrm{v}}_{\mathrm{ib}}$) according to our secret key S: if S(j) equals to 0 then we set both $\hat{\mathrm{v}}_{\mathrm{ia}}[\mathrm{j}]$ and $\hat{\mathrm{v}}_{\mathrm{ib}}[\mathrm{j}]$ to be $\hat{\mathrm{v}}_\mathrm{i}[\mathrm{j}]$, and if S(j) equals to 1, then then ($\hat{\mathrm{v}}_{\mathrm{ia}}\mathrm{j}$, $\hat{\mathrm{v}}_{\mathrm{ib}}\mathrm{j}$) will be two random values with the sum of $\hat{\mathrm{v}}$ (j). Then we produce the encrypted vector $\ddot{\mathrm{v}}_\mathrm{i} = (\mathrm{M_1^T}\ \hat{\mathrm{v}}_{\mathrm{ia}}\ , \mathrm{M_2^T}\ \hat{\mathrm{v}}_{\mathrm{ib}})$

- $TR \leftarrow TradoorGen(K, m_q)$. The data user will run the $TradoorGen$ to generate the trapdoor for his query image m_q to retrieve similar images from the cloud server and it should not leak any information to the cloud server about the query image or the results, $TradoorGen$ will be described as:

1- Produce the aggregated feature vector VLAD for the query images v_q from the m_q.

2- Alter the query feature vector $v_q = (v_1, v_2, \ldots, v_l)$ into $\hat{v}_q = \left(-2v_{q1}, -2v_{q2}, \ldots, -2v_{ql}, 1\right)^T$, then divide $\hat{v}_q$ into two random vector ($\hat{v}_{qa}$, $\hat{v}_{qa}$) according to our secret key S: if $S(j)$ equals to 1 then ($\hat{v}_{qa}[j]$, $\hat{v}_{qa}[j]$) will be equal to $\hat{v}_q$, and if $S(j)$ equals to 0, then ($\hat{v}_{qa}[j]$, $\hat{v}_{qa}[j]$) will be two random values with the sum of $\hat{v}$[j], then we produce the encrypted query vector $\ddot{v}_q = (\delta M_1^{-1}\ \hat{v}_{qa}\ , \delta M_2^{-1}\ \hat{v}_{qb})$, where δ is a real random positive value. The vector $\ddot{v}_q$ represents the trapdoor TR.

- $\phi \leftarrow Search(I, TR, C)$. When the cloud server receives the trapdoor TR from the data user, the $Search$ algorithm will be run to retrieve images similar to the query in the encryption domain. Since figuring out the nearest images is time-consuming in the search step, we employed the index tree to reduce the search time by recursively matching the query vector against only the most similar centroid nodes along the path from the root nodes to the leave nodes of the tree. When we find the most similar cluster, we compare our query trapdoor against all of its descriptors. This method significantly decreases the computation overhead. The distance between each descriptor vector and the query vector will be calculated and ranked according to the similarity scores. Only the top-ϕ similar images will be returned to the data user. Calculating the distance in the encryption domain will be as follows:

$$\begin{aligned} v_q'^T v_i' &= \left(\delta M_1^{-1}\hat{v}_{qa}\right)^T M_1^T \hat{v}_{ia} + \left(\delta M_2^{-1}\hat{v}_{qb}\right)^T M_2^T \hat{v}_{ib} \\ &= \delta\left(\hat{v}_{qa}\right)^T \hat{v}_{ia} + \gamma\left(\hat{v}_{qb}\right)^T \widehat{v_{ib}} \\ &= \delta\left(\hat{v}_q\right)^T \hat{v}_i \\ &= \delta\left(\|v_i\|^2 - 2\sum_{j=1}^{l} v_{i,j} v_{q,j}\right) \\ &= \delta\left(\|v_q - v_i\|^2 - \|v_q\|^2\right). \end{aligned} \quad (2)$$

We employ δ and $\|v_q\|^2$ to hide the distance $\left|\left|v_q - v_i\right|\right|^2$, if $v_q'^T v_1' > v_q'^T v_2'$ then $\|v_q - v_1\|^2 > \|v_q - v_2\|^2$, by sorting the set of the products $v_q'^T v_i'$ the cloud server will have the ability for to find the closest feature vectors without revealing the original aggregated feature vectors v. The final step is to send the top-ϕ similar encrypted images to the data user.

C. Security Analysis

In this part we will discuss the security issue of our proposed scheme.

1- *Image content privacy*: images could be encrypted with any standard method for data encryption. Thus, we will not consider its security as these methods are well defined and proved.

2- *Aggregated features privacy:* Recall that aggregated feature vectors are protected by ASPE method [11] which is proved to be secure against ciphertext-only attacks.
3- *Query trapdoor privacy:* the query image trapdoors are generated and encrypted by the same method for aggregated image vectors. Thus they are all bell protected too.
4- *Access and search pattern:* similar to previous PP-CBIR schemes, our scheme leaks the access and search pattern to the cloud server. Such information can be protected but at the expense of more computation and communication costs.

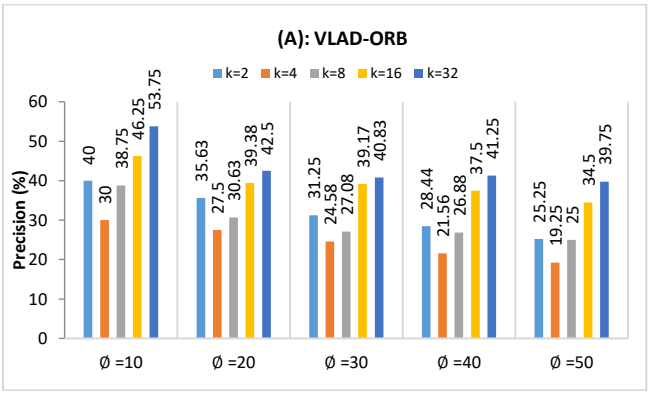
V. Experimental Results

Our scheme was implemented on a i5-7300U CPU @ 2.67GHz 2.71 GHz have two core and 4 logical processors, 8GB of RAM, and a hard disk 256 GB SSD, Windows 10 64-bit as an OS. Our scheme was released using MATLAB R2017b. The vector of locally aggregated descriptors (VLAD) was implemented using Python 3.9. Our experiments were performed on the real dataset Corel-1k, which includes ten categories with 100 images (with two resolutions of 384 × 256 pixels or 256 × 384 pixels) per one. Fig. 4 shows seven samples for each category. The performance of our scheme depends on the underlying local features, which are generated per image as sets of features of size d.

Fig. 4: Categories of Corel-1k dataset.

A. Retrieval Effectiveness

During our experiments, we used the precision metric to measure the retrieval effectiveness, which is defined as $Pr = \acute{\emptyset}/\emptyset$, $\acute{\emptyset}$ representing the real number of the relevant images that are retrieved. Notice that, the similarity equation (2) will be conducted over encrypted vectors without affecting the precision. We employed two local feature descriptors: SIFT[36] and ORB [8] feature vectors of sizes 128 and 32, respectively.
To test the retrieval precision, we submitted 20 image queries from the ten different categories. Therefore, the retrieval precisions are the average values of 20 search queries. Fig. 5 shows the average retrieval precision for different $\emptyset$ values. Recall that the aggregated vectors VLAD are generated by aggregating the local descriptors into k visual words. Our experiments are conducted for different k values: 2, 4, 8, 16, 32. Notice that SIFT descriptors are slightly better than ORB descriptors.



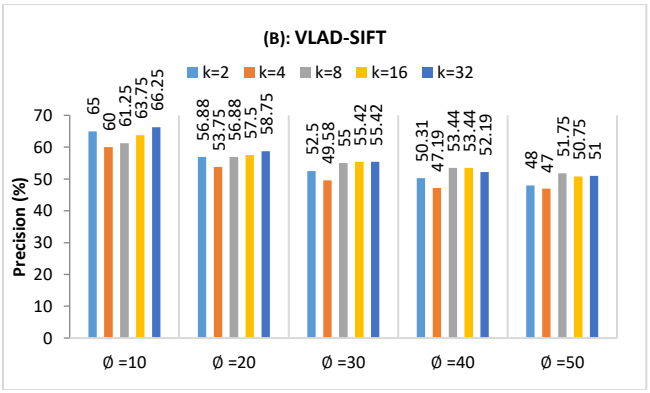


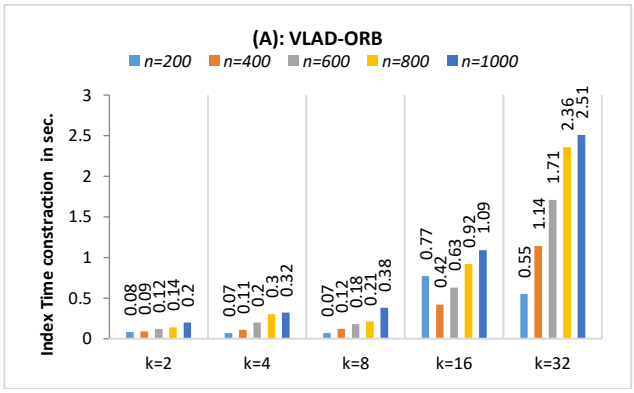
Fig. 5: Average precision. (A) ORB descriptors, (B) SIFT descriptors.

B. Efficiency Investigation

In this subsection, we investigate the efficiency of our scheme in terms of time consumption and storage cost. The time consumption is presented according to the index creation, trapdoor generation and search operation.

1- *Index construction time***.** Recall that the secure index is constructed as a tree index from the aggregated features for the entire image collection. Each aggregated feature is generated from a set of (k) visual words derived from the local features of the corresponding image. Before being outsourced, the entire index is encrypted using ASPE method**,** which requires a break apart operation and pair of multiplicative operations with the $(l+1) \times (l+1)$ matrices. The time complexity of the break apart is $O(nl)$, and the time complexity for the multiplication for the matrix is $O(nl^2)$. So, the encryption complexity is $O(nl + nl^2)$. Fig. 6 illustrates the index construction times for a variable number of images n. Notice that ORB local descriptors consume less time than SIFT descriptors since the size of ORB descriptors is shorter than that of SIFT. Obviously, a greater number of images requires more indexing time.
2- Trapdoor Generation Time
Search trapdoors are generated and encrypted in the same method for image aggregated vectors. Trapdoor encryption incurs a splitting operation, and two matrix multiplications. Thus, its complexity is $O(l + l^2)$. Fig. 7 reports the trapdoor generation time for different local descriptors.

Again, ORB consumes little time as it has low dimensional descriptors.



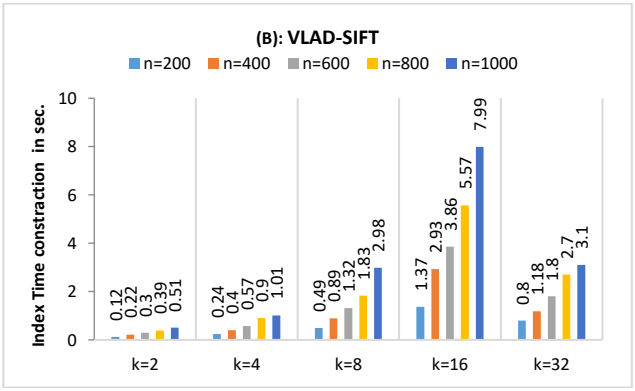
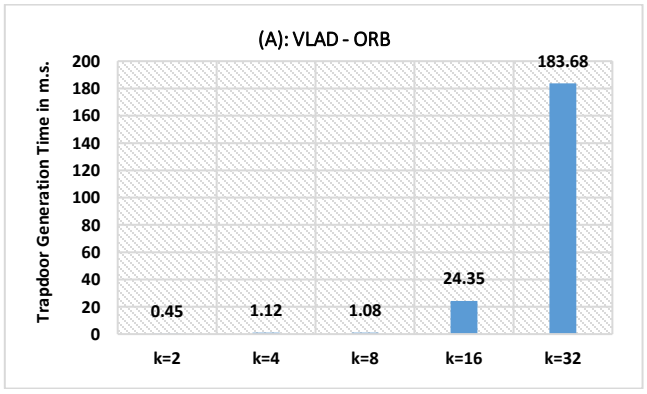


Fig. 6: The time cost of secure index construction. (A): ORB descriptors, (B): SIFT descriptors.



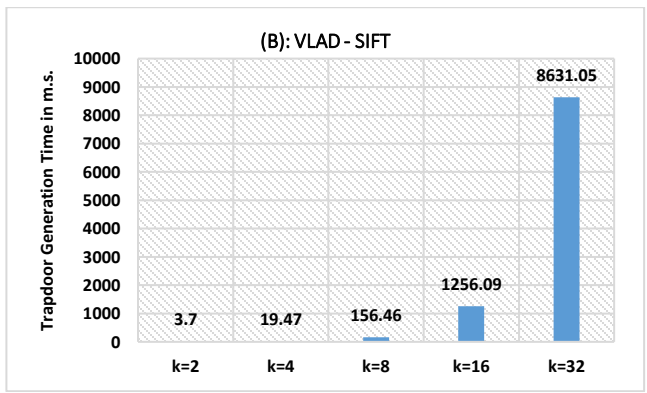
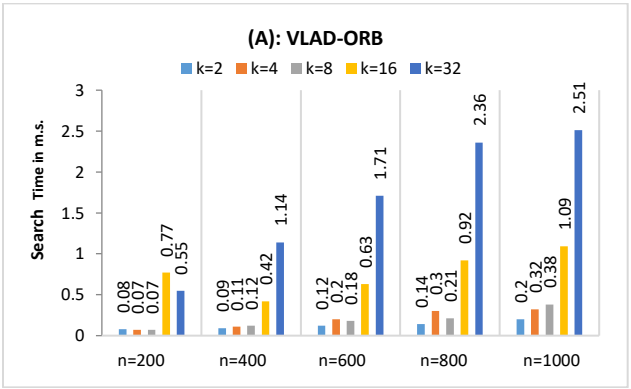


Fig. 7: The time cost of trapdoor generation. A): ORB descriptors, (B): SIFT descriptors.

3- Search time: Once the search request TR is provided, the cloud server first matched it with the tree-index to get the nearest class. Then, it matches TR against all the n' aggregated vectors within that class to retrieve the top-$\emptyset$ similar results. Thus the search time complexity of our scheme is O(n'). Notice that the value of n' is commonly significantly smaller than the total number of images *n*. Fig. 8 illustrates the search time for a variable number of images with different variations of aggregated vectors. Notice that ORB-variants are similar to SIFT-variants since they are described into the same form.



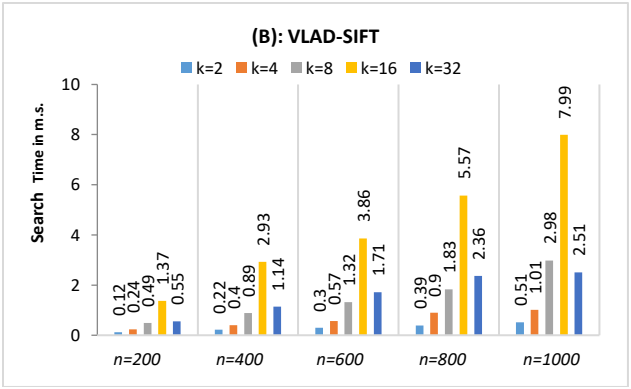


Fig. 8: Time cost for relevant images search in an encrypted dataset containing 1k images. . A): ORB descriptors, (B): SIFT descriptors.

4- *Storage cost:* Fig. 9 (A ,B) illustrates the storage cost of the secure index with two descriptors for variable number of visual words. The results are obtained by indexing 1000 images. In addition shows that ORB descriptors consume a little storage compared with the SIFT descriptors. Also, more visual words incur more storage costs.

5- *Round trip cost*. Our proposed scheme requires only one round to retrieve similar images without any communication between the data owner and user. This is because of utilizing ASPE for protecting the trapdoor search

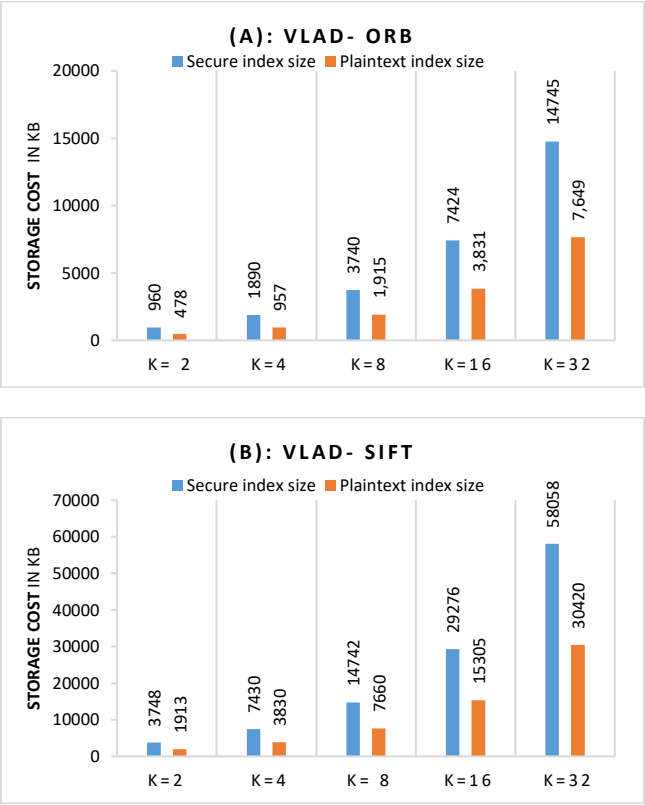


Fig. 9: Storage requirements of the secure index & Plaintext index in (KB) for Corel-1k. A): ORB descriptors, (B): SIFT descriptors.

VI. Conclusions

In Our paper, we designed and apply a new PP-CBIR scheme within the setting of the cloud computing. Each image is described as a single compact aggregated vector that is derived from local descriptors. This method significantly reduces the computation and commination costs. The aggregated vectors are encrypted using ASPE algorithm, which enables the cloud server to calculate the resemblance scores for the encrypted image feature vectors without decryption or any additional round of communication. The image feature vectors are indexed as tree-index to improve the search efficiency from $O(n)$ to $O(n')$. Our experiments are performed for two popular local descriptors: ORB and SIFT. The aggregated vectors are generated with a variable number of visual words. Results illustrate the practical value of our proposed scheme. For future work, we try to embed invisible watermarks for preventing dishonest users from the illegal distribution of images.

Conflict of Interest

The authors have no conflict of relevant interest to this article.

References

[1] Y. Mingqiang, K. Kidiyo, and R. Joseph, “A survey of shape feature extraction techniques,” *Pattern recognition*, vol. 15, no. 7, pp. 43-90, 2008.

[2] H. Al-Jubouri, and H. Du, “A Content-Based Image Retrieval Method By Exploiting Cluster Shapes,” *Iraqi Journal for Electrical & Electronic Engineering*, vol. 14, no. 2, 2018.

[3] J. R. Smith, and S.-F. Chang, "Tools and techniques for color image retrieval", *Electronic Imaging*, pp. 426-437, 1993. DOI:10.1117/12.234781

[4] B. S. Manjunath, and W.-Y. Ma, “Texture features for browsing and retrieval of image data,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 8, pp. 837-842, 1996.

[5] S. K. Abdulateef, and M. D. Salman, “A Comprehensive Review of Image Segmentation Techniques,” *Iraqi Journal for Electrical & Electronic Engineering*, vol. 17, no. 2, 2021.

[6] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91-110, 2004.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Computer vision and image understanding*, vol. 110, no. 3, pp. 346-359, 2008.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF", 2011 International Conference on Computer Vision. pp. 2564-2571, 2011.

[9] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971-987, 2002.

[10] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation", 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3304-3311, 2010.

[11] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases", Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, pp. 139–152, 2009. DOI: 10.1145/1559845.1559862

[12] Z. A. Abduljabbar, A. Ibrahim, M. A. Hussain, Z. A. Hussien, M. A. Al Sibahee, and S. Lu, “EEIRI: efficient encrypted image retrieval in IoT-cloud,” *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 13, no. 11, pp. 5692-5716, 2019.

[13] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, "Enabling search over encrypted multimedia databases", Proceedings Volume 7254, Media Forensics and Security, 725418, 2009. DOI:10.1117/12.806980

[14] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu, "Secure image retrieval through feature protection", 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1533-1536, 2009.

[15] L. Zhang, T. Jung, K. Liu, X.-Y. Li, X. Ding, J. Gu, and Y. Liu, “Pic: Enable large-scale privacy preserving content-based image search on cloud,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 11, pp. 3258-3271, 2017.

[16] W. 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.
[17] A. I. Abdulsada, A. N. M. Ali, Z. A. Abduljabbar, and H. S. Hashim, "Secure image retrieval over untrusted cloud servers," *International Journal of Engineering and Advanced Technology*, vol. 3, no. 1, pp. 140-147, 2013.
[18] J. Qin, H. Li, X. Xiang, Y. Tan, W. Pan, W. Ma, and N. N. Xiong, "An encrypted image retrieval method based on Harris corner optimization and LSH in cloud computing," *IEEE Access*, vol. 7, pp. 24626-24633, 2019.
[19] 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 transactions on information forensics and security*, vol. 11, no. 11, pp. 2594-2608, 2016.
[20] Z. Xia, Y. Zhu, X. Sun, Z. Qin, and K. Ren, "Towards privacy-preserving content-based image retrieval in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 276-286, 2015.
[21] H. Cheng, X. Zhang, J. Yu, and Y. Zhang, "Encrypted JPEG image retrieval using block-wise feature comparison," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 111-117, 2016.
[22] H. Cheng, X. Zhang, J. Yu, and F. Li, "Markov process-based retrieval for encrypted JPEG images," *EURASIP Journal on Information Security*, vol. 2016, no. 1, pp. 1-9, 2016.
[23] B. Ferreira, J. Rodrigues, J. Leitao, and H. Domingos, "Privacy-preserving content-based image retrieval in the cloud", 2015 IEEE 34th Symposium on Reliable Distributed Systems (SRDS), pp. 11-20, 2015.
[24] 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, 2017.
[25] H. Wang, Z. Xia, J. Fei, and F. Xiao, "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. Lu, T. Qin, H. Shim, X. Chen, and B. Jeon, "A privacy-preserving image retrieval based on AC-coefficients and color histograms in cloud environment," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 58, no. 1, pp. 27-43, 2019.
[27] 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 Transactions on Services Computing*, 2019.
[28] Z. Xia, X. Ma, Z. Shen, X. Sun, N. N. Xiong, and B. Jeon, "Secure image LBP feature extraction in cloud-based smart campus," *IEEE Access*, vol. 6, pp. 30392-30401, 2018.
[29] 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," *Journal of Visual Communication and Image Representation*, vol. 43, pp. 164-172, 2017.
[30] J. Anju, and R. Shreelekshmi, "Secure content-based image retrieval using combined features in cloud", International Conference on Distributed Computing and Internet Technology (ICDCIT 2020): Distributed Computing and Internet Technology, pp. 179-197, 2020.
[31] Z. Qin, J. Yan, K. Ren, C. W. Chen, and C. Wang, "Towards efficient privacy-preserving image feature extraction in cloud computing", Proceedings of the 22nd ACM international conference on Multimedia, pp. 497–506, 2014. DOI:10.1145/2647868.2654941
[32] J. Anju, and R. Shreelekshmi, "A faster secure content-based image retrieval using clustering for cloud," *Expert Systems with Applications*, vol. 189, pp. 116070, 2022.
[33] Q. Gu, Z. Xia, and X. Sun, "MSPPIR: Multi-source privacy-preserving image retrieval in cloud computing," *arXiv preprint arXiv:2007.12416*, 2020.
[34] J. Sivic, and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos", Proceedings Ninth IEEE International Conference on Computer Vision, pp. 1470-1470, 2003.
[35] C. Gentry, "Fully homomorphic encryption using ideal lattices", Proceedings of the forty-first annual ACM symposium on Theory of computing, pp. 169–178, 2009. DOI:10.1145/1536414.1536440
[36] C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, "Secure and robust SIFT", Proceedings of the 17th ACM international conference on Multimedia, pp. 637–640, 2009. DOI:10.1145/1631272.1631376