Received: 29 May 2022 | Revised: 03 July 2022 | Accepted: 03 July 2022
DOI: 10.37917/ijeee.18.2.11 Vol. 18| Issue 2| December 2022

Open Access

Iraqi Journal for Electrical and Electronic Engineering
Original Article

Secure Content-Based Image Retrieval with Copyright Protection within Cloud Computing Environment

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
Every day, a tremendous amount of image data is generated as a result of recent advances in imaging and computing technology. Several content-based image retrieval (CBIR) approaches have been introduced for searching image collections. These methods, however, involve greater computing and storage resources. Cloud servers can address this issue by offering a large amount of computational power at a low cost. However, cloud servers are not completely trustworthy, and data owners are concerned about the privacy of their personal information. In this research, we propose and implement a secure CBIR (SCBIR) strategy for searching and retrieving cipher text image databases. In the proposed scheme, the extract aggregated feature vectors to represent the related image collection and use a safe Asymmetric Scalar-Product-Preserving Encryption (ASPE) approach to encrypt these vectors while still allowing for similarity computation. To improve search time, all encrypted features are recursively clustered using the k-means method to create a tree index. The results reveal that SCBIR is faster at indexing and retrieving than earlier systems, with superior retrieval precision and scalability. In addition, our paper introduces the watermark to discover any illegal distributions of the images that are received by unlawful data users. Particularly, the cloud server integrates a unique watermark directly into the encrypted images before sending them to the data users. As a result, if an unapproved image copy is revealed, the watermark can be extracted and the unauthorized data users who spread the image can be identified. The performance of the proposed scheme is proved, while its performance is demonstrated through experimental results.

KEYWORDS: Searchable Encryption, Content-Based Image Retrieval (CBIR), Asymmetric Scalar-Product-Preserving Encryption (ASPE), VLAD, Watermark, Copyright Protection

I. INTRODUCTION

Content-based image retrieval (CBIR) is a helpful approach for finding images in the image collections. Such an approach has been used for many years in a variety of real-world applications such as face recognition, object identification, and medical detection. The increased usage of digital cameras and cellphones has resulted in massive image archives. As a result, standard CBIR methods will be impractical since they need more storage and computational resources. Cloud computing can aid by giving data owners on-demand access to sufficient storage and large computational resources. CBIR can be used by authorized users to contact the cloud server and get similar results. This setting raises two privacy threads. The first one, cloud server will break the privacy of the data owners since images are no longer under the supervision of their owners. The second one, authorized users might spread the images obtained through unauthorized data users.

To mitigate such threads, images have to be encrypted before being transferred to cloud servers. Unfortunately, CBIR activities will be disabled if traditional encryption techniques are used directly. As a result, developing secure CBIR systems (SCBIR) that can deal with encrypted images without decryption is utmost important.

In the following demonstrate how the existing secure CBIR schemes work: the data owner extracts some feature vectors from each image. Then, before being sent to the cloud server, all images and vectors are encrypted. In this case, the distance between two encrypted vectors can be used to determine the similarity between their corresponding images. Image feature vectors can be either global, which creates a summary vector for the entire image, or local, which represents the image by its interest spots, resulting in a large number of feature vectors. Global features, on the other hand, are dependent on the image's signal representation; any change in illumination, scaling, rotation, or color depth in the same image will result in a new 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 .On another hand, the local feature vectors 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 a single pixel but it will make use of its neighbors pixels .The most well-known local feature image descriptors are SIFT [6], SURF[7], ORB[8], and LBP[9] . Each image descriptor will have its own length; SIFT has $d = 128$ dimensional with positive values. Many applications, such as image search and identification, use local image descriptor representations. This type of encoding, on the other hand, will result in a large number of feature vectors for the single image. As a result, image similarity will need a lot of processing power. As a result, numerous strategies for dealing with the large number of feature vectors have been presented. Particularly, in this paper the local feature descriptors aggregations approach VLAD[10] was used with two local descriptors as our mean image descriptors: (ORB) and (SIFT).

Some SCBIR schemes use homomorphic encryption (HE) [11] to protect the aggregated vectors, which allows some arithmetic operations on the encrypted data. However, HE entails a great amount of complexity. Instead, also used Wong et al ASPE [12] method. This scheme has the ability to compute secure kNN [12] similarity of two encrypted vectors without being decrypted. However, to compare the provided query against the current encrypted vectors, the cloud server must perform a large number of operations. To address this problem, also use a hierarchal-indexing method based on the k-means clustering algorithm to improve search efficiency. The encryption key must be shared with authorized data users who create the trapdoor for their query image. The data user privacy is protected in this setting because the data owner has no idea what the user is looking for.

Our Contributions. The summarize of our contributions with the following points.

1. Using the VLAD approach with combine all the local feature descriptors into one aggregated feature vector. Fast searching and indexing will be possible with this technique.
2. To safeguard the aggregated vectors with the use a mild encryption approach called ASPE, which is scalable and efficient.
3. In this paper k-means technique used for clustering to create a hierarchal index to improve search efficiency.
4. Two well-known local descriptors used in this paper: ORB and SIFT,
5. Watermarking technology implemented to protect the images from the unauthorized distribution

Paper Organization. The rest of this paper is divided as follows: The existing SCBIR schemes are shown in Section II. A brief background is given in Section III. Our proposed scheme is presented in section IV. The results are reported in Section V. The paper conclusion is provided in Section VI.

II. Related Works

Current SCBIR methods operate in one of two modes: encrypted features schemes or encrypted image schemes. In the first mode, the data owner extracts image features and encrypt them before storing in the cloud, whereas in the second mode, the data owner encrypts images and delegates the feature extraction to the cloud server.

A. *Encrypted features schemes*

Several SCBIR designs have been proposed in recent years. Lu, et al. [13] proposed a privacy-preserving CBIR, where images are defined as global histograms of visual words. Encryption with Order preserving, or min-hash algorithms are used to encrypt such histograms. To determine the similarity between two images, The distance between their histograms was calculated using the Jaccard distance. Lu, et al. [13] Encrypts color histograms using bit randomization, random projection, and random unary encoding techniques. All three approaches, however, have showed low retrieval accuracy. To achieve better accuracy, several researchers [14] [15] used homomorphic encryption. In the case of HE, however, calculating the distance between the encrypted query vector and the encrypted outsourcing vectors requires an active communication between the cloud server and the data owner. Qin et al [16] suggested a secure image retrieval system based on Harris' corner descriptor and using the locality-sensitive hash (LSH) method, which has a moderate retrieval accuracy .Abduljabbar, et al. [17] he suggested the image will be represented in local features measuring it with Euclidean distance . [18] suggest using ASPE to measure similarity of global feature vectors. Xia, et al. [19] constructed a secure CBIR, where images are described using SIFT local features within the Bag of Visual Words (BOVW) model. Herein, the distance between two images is measured using the Earth Mover's Distance (EMD). Such a solution requires multiple interactions between the data owner and the cloud provider in order to find images that are equal to the query image.

B. *Encrypted image schemes*

Cheng, et al. [20] presented a secure CBIR method for JPEG images, where the encrypted image is decrypted, and five descriptive elements are retrieved. The retrieval accuracy of [20] is further improved in [21]. Ferreira, et al. [22],[23] proposed to encrypt images by substituting image pixels and shuffling the results. Then a global histogram was formed from the encrypted images. Wang, et al. [24] developed a method for extracting random features from AES-encrypted images with low retrieval accuracy. Xia, et al. [25] has proposed a method for encrypting images in the YUV color space., where two histograms are extracted and then concatenated from this encrypted image. The Manhattan approach is used to evaluate the distance between two images vectors. Despite this, the search accuracy in this strategy was embarrassingly low. Xia et al [26] have utilized BOVW framework to extract the feature of the encrypted image. Their scheme has a security flaw since it employs weak encryption primitives. Xia et al. [9] have combined pixel and block shuffling to create a secure Local Binary Pattern (LBP) feature. However, none of the current schemes

take into account dishonest data users s who may be acting illegally by distributing retrieved images for unauthorized users. Preventing image distribution is a difficult task. Instead, certain techniques can be devised to detect such illegal behavior. Watermarking technique has been extensively researched [20, 22, 27-32] for copyright protection in buyer-seller scenarios. The seller inserts a one-of-a-kind code to prevent copying. Before giving access to an image to a data user, a specific watermark is embedded within the image. If a data user distributes the watermarked image copies, data user will be identified by extracting the watermark. The following issues must be resolved in order to use watermark-based copyright protection:

- Watermark should be embedded in all the received images for each query request in order to prevent illegal copying. Such a process entails high computational complexity. So, its need to use an efficient watermarking technique that allows the cloud server to immediately embed the watermark in the images that has been encrypted. After receiving the encrypted and watermarked image, the data user should be able to directly extract the watermarked images using the given encryption information. The watermark is still there in the image after it has been decrypted.
- The data owner can accuse a data user in a watermark-based copyright protection scheme by watermarking an image with the user's watermark This type of illegal behavior must be avoided.
- Upon getting the watermarked images, the data user can manipulate them using various image processing activities, such as JPEG compression, before sharing them illegally. Throughout scenario, the fingerprint bits could not be recovered with 100 percent precision. As a result, the trail with the extract faults must be carefully investigated.

III. Problem Description

A. System model

As seen in Fig. 1 : The proposed scheme SCBIR., our suggested architecture includes four key entities: the data owner , the authorized data users, the cloud service provider, and the watermark authenticator.

Data owner (DO) 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 DO extracts aggregated local feature vectors $V = (v_1, v_2, \dots, v_n)$ from the plaintext image collection, and then creates a secure index tree I from V. Then both C and I will be both stored in the cloud server. DO should authorize the data users via a specific authentication scheme, which is outside the scope of our work as many existing SCBIR schemes [9, 14, 18, 27, 33, 34] . Only the authorized users can submit valid search requests to the cloud server. In this paper, the consideration of the setting of a single DO. If multiple owners using the system with different sets of data users, then indexes, image collections, search requests are all encrypted with different keys.

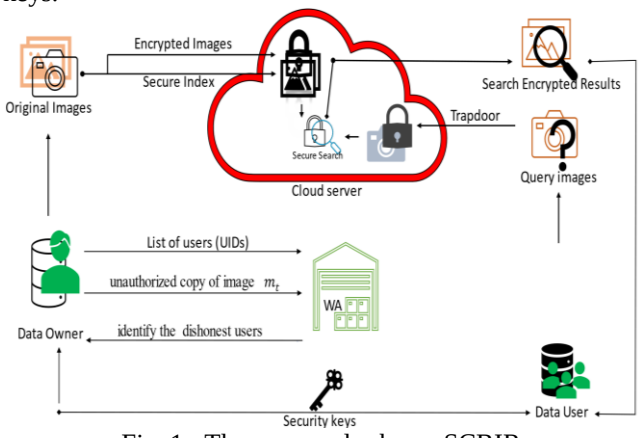

Fig. 1 : The proposed scheme SCBIR.

The data users (DU) are people who have been given permission by the DO to search the encrypted collection for query images. To access images, the DU must supply the cloud with a valid search trapdoor TR. When he/she receives the encrypted results, he/she will decrypt the encrypted results using the secret keys given by the DO.

The cloud server (CS) carries the burden for maintaining the encrypted image collection and its encrypted index, as well as the processing capacity required to respond to DUs' search queries.

The watermark authenticator (WA) is a legitimate source that generates watermarks for approved DUs and makes decisions after the results of its watermark extraction algorithm.

B. Design Goals

The following goals are targeted in this paper:

1. *Efficiency.* For large image collections, linear search is completely pointless and expensive by default. To improve search performance, this paper suggests using a secure tree index.
2. *Data privacy* the image collection's real information, image characteristics, and search queries should all be safely shielded from the semi-trusted CS.
3. *Protecting the copyright* this paper considers a semi-trusted DU who may distribute the returned images to other persons. Our proposed scheme utilizes the watermarking technique to prevent such an illegal distribution. Furthermore, the DO may accuse innocent DUs by fabricating their own watermarks in the original image. Such a behavior should be prevented in our scheme.

C. Thread model

This appear follow the previous SCBIR schemes [23, 27, 30, 35] to treat CS as a *semi-honest* entity, meaning that it follows and implements the protocol but tries to extract additional secret information from the communication data. Therefore, the image collection, secure index, and search trapdoors must be properly guarded.

This paper assumes that there is no collusion among the four parties of our proposed scheme. This assumption will allow for creating an efficient scheme. DO will not be able to obtain the watermarks, embedding method, or secret keys from either CS or the WA as a result of this. Furthermore, any secret keys will not be leaked to CS from the DUs. Similar to prior SCBIR techniques, [22, 26, 27, 30, 33], our scheme leaks to CS: the identities of returned images (access pattern), and whether the same image query has been searched before or not (search pattern).

D. *Vector of locally aggregated descriptors*

To cope with big image collections, local features must be optimized. Several quantizing techniques (aggregating) have been established to compact image characteristics into single descriptors vector without sacrificing accuracy. Bag-of-features (BOF) [36] and vector of locally aggregated descriptors (VLAD) [10] are most known methods .

BOF transforms the local descriptors $f_i \in R^d$ of a given image into a single histogram of size k, where k is *the number of* centroids $\theta = (\theta_1, \theta_2, \dots, c\theta_k)$ obtained using the k-means algorithm over the whole local descriptors for entire collection. However, BOF quantizes the local descriptor to its nearest visual world regardless of quantization error. VLAD, on the other hand, will keep track of all the differences between the descriptors while reducing the amount of the quantization errors that affects the outcome. 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. 2 illustrates the VLAD vector for similar images.

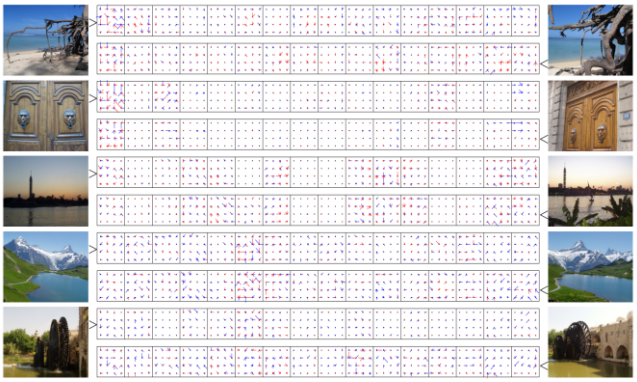

Fig. 2: VLAD descriptors, for $k = 16$ centroids, $d = 128$.

IV. Proposed Scheme

A. *Overview of the proposed scheme*

The proposed scheme includes several algorithms that work as follows: DO runs $KeyGen$ over the security parameter λ to obtain the secret key set K. He runs the $ImgEnc$ and $IndexGen$ algorithms over the image collection M and K to get the encrypted images C and the secure index I, respectively. The values of C and I are stored in CS and the set K is shared with the authorized users. Furthermore, the DO provides WA with the set of legal user identities UID_i. WA runs the $WatermarkGen$ to create a one-of-a-kind watermark w for each user in UID_i, The watermarks w are then sent to CS by WA. When an authorized DU wants to retrieve similar images to his query image q, DU runs $TradoorGen$ over q and K to generate the secure trapdoor TR, which is submitted to CS. The CS runs $Search$ over TR and I and returns the top-ϕ most similar encrypted images R. Upon receiving R, CS locates the watermark w corresponding to the current DU and uses $WatermarkEmb$ to embed w into each of the images in R obtaining the watermarked image set R'. When DU receives R'he/she executes $ImgDec$ to get the set of decrypted images M_q. Please keep in mind that the watermark is still remain in the decrypted images. When DO discovers that any DU unlawfully distributes an image M_t in M_q then DO will submit M_t and its original version M_o to WA. WA then uses $WatermarkExtra$ to extract the watermark W_t from M_t and determines its corresponding DU who is linked with the watermark W_t.

Below will present our scheme for secure content-based image retrieval in more details. For better understanding, also will divide the presentation of our work into two parts: the first part will consider the basic secure CBIR, while the second part will deal with the copyright protection.

B. *Secure CBIR scheme*

This subsection will explain the algorithms of DO side ($KeyGen$, $IndexGen$), DU side ($TradoorGen$, $ImgDec$), and CS side ($Search$) in more details.

- $K \leftarrow KeyGen(\lambda)$ The security parameter λ will be received by the algorithm 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.
- $(I) \leftarrow IndexGen(K, M)$ the secure index I is returned by this algorithm, which requires K and M as inputs the secure index is generated by the following steps.

1- *Index generation*

For each image $m_i \in M$ the DO extracts the set of local feature vectors (descriptors) $F = (f_1, f_2, \dots, f_z)$, 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 to conduct the search when the implement a large number of images, however, this one-to-one index will be impractical. the proposed scheme created a tree-index to help speed up the search. The k-means clustering algorithm is used to create this index, with k being a user-defined variable that represents the number of centroids that describe the entire image vectors. Also, build a tree index recursively using a k-means clustering algorithm to reduce searching time. Images (aggregated descriptors)

within the same cluster are likely to be similar in this context, and then can quickly discard dissimilar images that are far from the query image during the search. During the search, the query image will only be compared recursively with the closest centers. Finally, CS will compute similarity scores for all remaining images in the cluster. The computation cost is significantly reduced as a result of this treatment.

2- Index encryption

Image descriptors may reveal some information about its real content. As a result, before sending the aggregated descriptors to CS, DO must encrypt them. The encryption method should allow for the ranking and retrieval of encrypted descriptors without the need for decryption. Homomorphic encryption [11] is commonly utilized to accomplish this purpose, perhaps at the cost of increased searching time and communication loops between the data consumer and CS. Alternatively, the ASPE algorithm [12] is used to protect the aggregated descriptors.

To do so, first extend the aggregated image feature vector $\mathrm{v_i} = (\mathrm{v_{i,1}}, \dots., \mathrm{v_{i,l}})^{\mathrm{T}}$ into $\hat{\mathrm{v}}_{\mathrm{i}} = \left(\mathrm{v_{i,1}}, \mathrm{v_{i,2}}, \dots, \mathrm{v_{i,l}}, \left|\left|\mathrm{v}\right|\right|^2\right)^{\mathrm{T}}$ where $\left|\left|\mathrm{v_{i,l}}\right|\right|$ is the Euclidean norm of $\mathrm{v_i}$, then 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 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 produce the encrypted vector $\ddot{\mathrm{v}}_{\mathrm{i}} = (\mathrm{M_1^T}\ \hat{\mathrm{v}}_{\mathrm{ia}}\ , \mathrm{M_2^T}\ \hat{\mathrm{v}}_{\mathrm{ib}})$

- $TR \leftarrow TradoorGen(K, m_q)$. DU would use $TradoorGen$ to get the trapdoor TR for his search query M_q in order to retrieve similar images from CS. TR should not leak any details about the query image or the outcomes towards CS. $TradoorGen$ will be discussed in the following fashion:

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

2- Alter the query feature vector $v_q = (v_1, v_2, \dots, v_l)$ into $\hat{v}_q = \left(-2v_{q1}, -2v_{q2}, \dots, -2v_{ql}, 1\right)^T$, 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 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 CS receives TR from DU, it runs $Search$ algorithm to retrieve images in the encryption domain that are similar to the query. Because determining the closest images in the search step takes additional time, in addition, using the index tree to speed up the process by recursively matching the query vector against only the most similar centroid nodes along the tree's path from the root nodes to the leaf nodes. The proposed scheme compares our query trapdoor against all of the cluster's descriptors once we've found the most similar cluster. The computation overhead is significantly reduced using this method. According to the similarity scores, the distance between each descriptor vector and the query vector will be calculated and ranked. Only the top-ϕ similar images will be returned to DU. Calculating the distance in the encryption domain will be as follows:

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

The proposed scheme employ δ and $\|v_q\|^2$ to hide the distance $\left||v_q - v_i|\right|^2$, 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 DU. Finally, CS creates a temporary set R with the $top - k$ most comparable encrypted images. These images will be given to the inquiry user once they have been watermarked as R'.

C. Copyright protection

In the suggested scheme, the watermarking technology is employed to protect the images from the unauthorized distribution. The cloud server embeds a unique watermark w_i, connected with each data user DU_i, into the encrypted relevant images. Then the data user is supplied with the encrypted and watermarked images. Then he/she can immediately decrypt the images after obtaining them. Notice that, after decryption, the watermark is still exists. When an illegal replica m_t of an original image is detected, the data user who committed the illegal act can be identified by extracting his watermark from m_t.

Commonly, Homomorphic cryptosystems are used to build secure watermarking algorithms [37-40], where a new watermark is inserted into all the encrypted images and recovered from decrypted images. However, such schemes are not well suited to the application of image retrieval. This is because; Homomorphic encryption takes a very long time for each search query in group of images. Particularly, embedding a watermark in this case is a complex process. Alternately, creating, at the beginning, a unique watermark for each data user. In this case, multiple search results, will be embedded with the same watermark for a given data user.

In our scheme, the use a CEW algorithm provided by Zhang [41] for its security and efficiency. The embedding and extraction processes are adjusted in [41] to enhance the accuracy of watermark identification.

According to Zhang's approach [41], each pixel in a grayscale image is made up of 8 binary bits. An exclusive-Or procedure is used to encrypt the image's pixel bits into random bits. with a conventional stream cipher in Zhang's algorithm's embedding procedure. The encrypted image is then split into non-overlapping blocks. Some of these blocks are picked randomly to hold the watermark bits. Following that, according to a secret key, the pixels in each of the chosen blocks are split into two groups at random S_0 and S_1. The embedding process is performed as follows: Flip the firth two least significant bits ($LSBs$) of the pixels in S_0 if the bit of the watermark is 0; else flip the first two $LSBs$ of the pixels in S_1. The same stream cipher is generated to decrypt the encrypted and watermarked image. The watermark is still present in the encrypted image. Watermark bits are extracted based on the fact that the fluctuation of an original image block is generally smaller than that of a flipped image block. Watermark extraction processes as follows: a secret key is used to locate the blocks containing the watermark bits. Then, pixels of each block are separated according to a secret key into two groups: S_0 and S_1. Finally, the first two $LSBs$ pixels in S_0 and S_1 are flipped independently. Aslo may infer whether the embedded bit is 1 or 0 according to the fluctuation change. It should be noted that the extraction of each watermark bit is not guaranteed. Please see [41] for further details on Zhang's work.

The DO uses $ImgEnc$ to encrypt his M original images obtaining the encrypted image set C then outsourced C to CS after using a typical stream cipher. When CS receives the search request TR from the data user DU_i, it receives the temporary result of a search R and $w_i$as watermark related with DU_i and uses $WatermarkEmb$ to embed w_i into the images in R, resulting in the watermarked-encrypted image collection R', which is then delivered to DU_i as the final result set. The data user receives R' and decrypts the images in R' to obtain the plaintext images.

Suppose M_q is the decrypted and watermarked images. If an unauthorized distribution of image m_t is detected, The DO transmits WA the illegal copy m_t as well as the original version m_o , which uses $WatermarkExtra$ to get the watermark w_t. Finally, the extracted $w_t$is utilized to find the unauthorized user with a watermark that looks very similar to w_t. The suggested watermarking strategy is estimated to outperform [41] in terms of extraction speed. Algorithm 1 show the details of the watermark embedding and extraction algorithms. Keep in mind that these techniques were designed with monochrome images in mind. Colored images may be handled by merely repeating the RGB color operations.

Algorithm 1: The watermark embedding and extraction algorithm

$\mathcal{R}' \leftarrow$ WatermarkEmb $(\mathcal{R}, w, k_{emb1}, k_{emb2})$

1- the image that encrypted $c \in \mathcal{R}$

- Split c into $s \times s$ blocks that are not overlapped. The watermark w is a bit pattern that is indicated as $= w_1, w_2, \dots, w_{N_w}$. A set of blocks $\{BK_i\}_{i=1}^{N_w}$ are selected randomly using the secret key k_{emb1}.
- For each watermark bit $w_i, i \in [1, \dots, N_w]$,
  - using the secret key k_{emb2}, the pixels in block BK_i are divided to separated groups S_0 and S_1.If $w_i = 0$, flip the first LSB pixels of S_0. Otherwise, flips the pixels of S_1.

2- Create a group of images that are encrypted and watermarked $\mathcal{R}'$.

$w_t \leftarrow WatermarkExtra\ (m_t, m_o, k_{emb1}, k_{emb2})$

- Use k_{emb1} to divide m_t into $s \times s$ sections $\{BK_i\}_{i=1}^{N_w}$ without overlap.
- w_t=[]

3- For each $i \in [1, N_w]$,

- Separate the pixels of BK_i into two sets S_0 and S_1 according to k_{emb2}.
- Flip the first two LSB pixels in S_0 and $S_1$to get two blocks BK_i^0 and BK_i^1.
- Construct the corresponding block BK_i from the original image m_o with the same secret keys.
- Calculate $\delta_0 = \sum_{p_j \in BK_i, p_j^0 \in BK_i^0} \left(p_j^0 - p_j\right)^2$ and $\delta_1 = \sum_{p_j \in BK_i, p_j^1 \in BK_i^1} \left(p_j^1 - p_j\right)^2$.

- If $\delta_0 < \delta_1$,
  w_t=w_t||0.
- Else,
  w_t=w_t||1.

4- Output the watermark w_t.

D. Security Analysis

This part will discuss the security issues of our proposed scheme.

1. Data privacy

This includes the following issues:

I. *Image content privacy:* Any standard data encryption method could be used to encrypt images. As a result, our paper won't worry about its security because these methods are well-defined and proven [2, 17, 18, 22, 23, 27, 30].

II. *Aggregated features privacy:* Keep in mind that the ASPE method [12],which has been shown to be secure against ciphertext-only attacks as proven in [12], and it is used to protects aggregated feature vectors.

III. *Query trapdoor privacy:* For aggregated image vectors, the query image trapdoors are generated and encrypted using the same method. As a result, they are all well-protected.

IV. *Access and search pattern*: Our scheme leaks the access and search pattern to CS, similar to previous SCBIR schemes. Such data can be safeguarded, but at the cost of increased computation and communication costs.

2- Copyright privacy

This paper assumed that DUs would correctly obey the protocol definition, but they may disclose the obtained images with someone who isn't supposed to see them. For the sake of advantages to prevent piracy, watermarking techniques are used the unlicensed distribution.

DO, on the other hand, may attempt to accuse innocent DUs by fabricating their own watermarks in the original image. This illicit activity should be stopped.

I. *Framing image users:* to do so, the DO must know the distinctive watermark for each user, the embedding technique, and the embedding secret keys. The trustworthy WA generates the watermarks and sends them to CS in our scheme. The watermarks and embedding keys are only known by CS and WA. The DO is kept safe about the watermarks and embedded cryptographic keys. under the assumption that there is no cooperation between DO, CS, and WA. Thus, the DO would be unable to frame the image users in this instance.

II. *Find and trace the unauthorized distributor:* If DO discovers that one of his/her images has been exposed to an unauthorized party; DO can submit the unlawful copy as well as the original image to WA. The watermark will be extracted from the suspicious image by WA. The retrieved watermark is then used to figure out which DU is unlawful. However, after you've obtained the watermarked images, DU can alter them using standard image processing procedures, such as JPEG compression. If so, the watermarks could not be retrieved with 100% accuracy. The similarity between two watermarks w_a and w_b of length $N_W$is defined as$\frac{N_s}{N_W}$, where N_s is the number of matching values in two watermarks that occur around the same time. As an example, consider the watermarks '0111110110' and '1100101011' the length of the watermarks are 10, and the bits of two watermarks are at the second, fifth, and ninth positions is similar, as a result, the similarity between the two watermarks is 0.3. The suggested technique is set up so that any two watermarks w_a and w_b have a similarity of less than ϑ .The watermark derived from an illegally disseminated image is denoted by w_e, while the watermark connected with DU is denoted by w_q. DU will be suspected if the similarity between w_e and w_q is more than $(\rho > \vartheta)$. The false negative occurs when the watermark-based system misses an unlawful DU. A false positive occurs when an innocent DU is mistakenly identified as an unlawful DU. The watermark information extraction process determines the likelihood of a false negative. Notice that the false negative will not occur if the number of correctly recovered watermark bits is more than $\rho \cdot N_W$. the probability of false negative probability is

$$P_\rho = 1 - \sum_{i=\rho N_w}^{N_w} C_{N_w}^{i} \gamma^{i} (1-\gamma)^{N_w - i} \qquad (3)$$

Where γ represents the probability of the correct extraction. In our scheme, when the number of wrongly obtained bits reaches a certain threshold, a false positive will result if more than $(\rho - \vartheta).\ N_w$. Notice that small ϑ denotes a low degree of similarity in the scheme's resulting in low false positive and false negative probability. A tiny ϑ, on the other hand, results in smaller number of identified watermarks that are now available. Our watermark embedding method is similar to other schemes in terms of upper bound and lower bound embedded bits. When ϑ is determined, then it's hard to determine the upper bound of allowed watermark embedded bits. Only the lower bounder for the number of watermarks could be estimated.

V. Experimental Results

Our scheme was implemented with $i5-7300U$ CPU with two cores, $8GB$ of RAM, and $256\,GB$ SSD hard drive, using Windows 10 64-bit. Codes are written using $MATLAB\ R2017b$. $Python$ 3.9 is used to create the vector of locally aggregated descriptors (VLAD). Our tests were carried out on the genuine dataset Corel-1k, which consists of ten categories with 100 images each (with two resolutions of $384x256$ pixels and $256x384$ pixels).

A. Retrieval Effectiveness

During our tests, the measured retrieval effectiveness using the precision metric as $Pr = \acute{\emptyset}/\emptyset$, $\acute{\emptyset}$ The number of relevant photos that are retrieved in real time. It's worth noting that the similarity equation (2) will be applied to encrypted vectors with no loss of precision. Two local feature descriptors were used: SIFT[6] and ORB [8] feature vectors of sizes 128 and 32, respectively.

The effectiveness of our scheme is measure using precision metric, which is the ration of the number of relevant images to the number of retrieved images. Fig. 3 shows the average precision of 20 queries for variable number of retrieved results $\emptyset$. Different k (the visual words of VLAD) values are used in our experiments. Notice that SIFT descriptors are better than ORB.

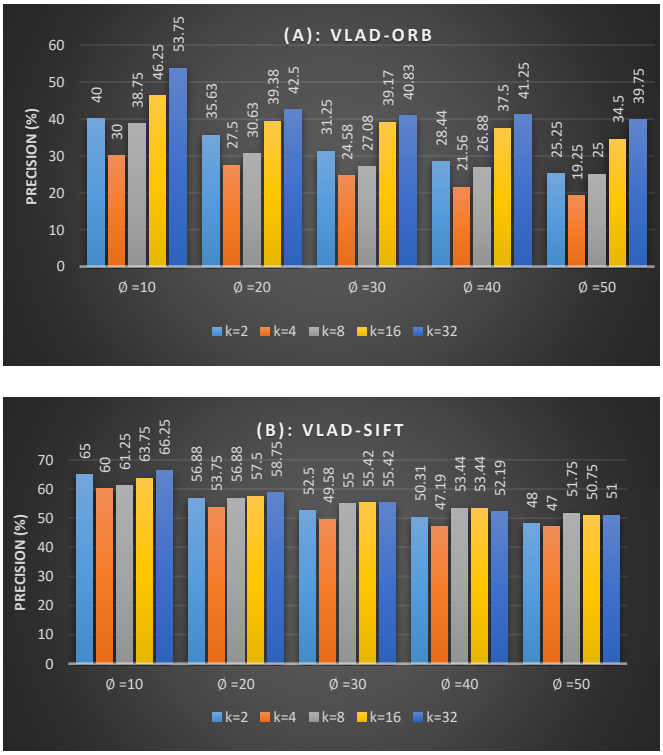

Fig. 3: Average precision. (A) ORB descriptors, (B) SIFT descriptors.

B. Index construction time.

Figure 4 shows the index construction times for a variable size of image collections. ORB descriptors require less time than SIFT, since it has smaller descriptors.

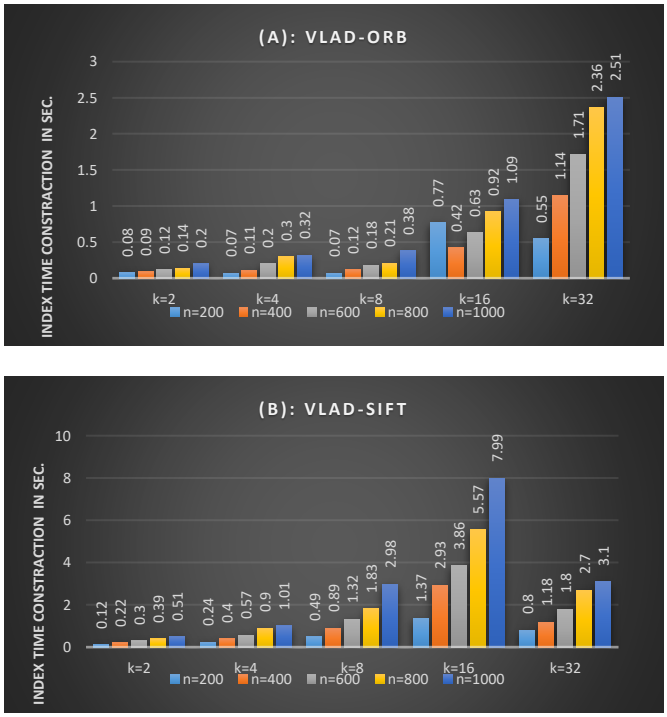

Fig. 4: The time cost of secure index construction. (A): ORB descriptors, (B): SIFT descriptors.

C. *Search time*

Fig. 5 shows the search time for a variable number of images with various aggregated vector variations.

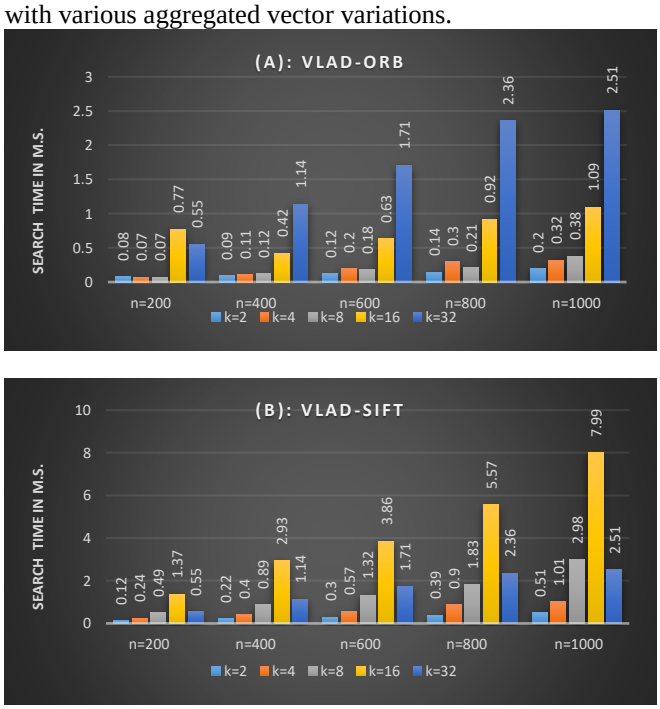

Fig. 5: Time cost for relevant images search in an encrypted dataset holding 1k images. A): ORB descriptors, (B): SIFT descriptors.

D. *Watermark extraction accuracy*

DU can modify the watermarked image using standard image processing activities, such like JPEG compression, after received it. Furthermore, if the user understands the watermarking process, by flipping the picture pixel bits, he/she might try to remove the watermark bits. In this case, it was impossible to obtain the watermark bits with 100% precision.

Notice that this work defines watermark extraction accuracy as $\gamma = \frac{N_e}{N_W}$, where N_W is the total of watermark bits and N_e is the number of successfully extracted watermark bits.

Please be aware that the embedding parameters used here may not be optimal. According to the application circumstances, the parameters may be adjusted freely to achieve a suitable balance between the robustness and retrieval quality. Table I shows the extractions for the embedded watermark without any attacks for 100 images.

TABLE I

Extraction accuracy γ without attack, where (psnr) is the watermarked image's peak signal to noise ratio, and (ssim) is structural similarity image measuring

Block size	PSNR (dB)	SSIM	Embedding time (s)	Extraction accuracy (γ) without attack
16	51.9785	0.9982	1.4613	100
24	48.3867	0.9962	4.5575	100
32	45.9136	0.9932	8.8241	100

To test the durability of the proposed watermarking technique, our study conducts bit-flipping and JPEG compress on watermarked image, and then attempt to extract watermark bits from these targeted images. The parameters used here are: N_W= 64, s = 16, 24, and 32.

With JPEG images, the overall extraction accuracy of the model is illustrated in Fig. 6, where images subjected to different quality factors QF. The results are based on 100 images. It is easy to see that bigger block sizes have better extraction accuracy and higher QF results in fewer extraction mistakes.

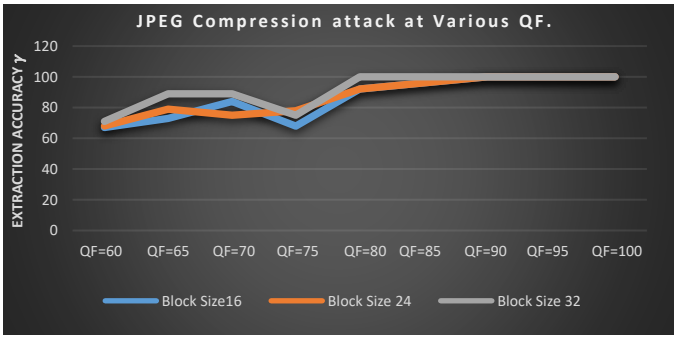

Fig. 6: Extraction Accuracy γ under JPEG Compression attack at Various Quality factors.

The overall extract accuracies are higher than 95% when 30% bits on two lower bit-planes are flipped randomly, as demonstrated in Fig. 7. However, image distortion cannot be ignored in this case. It implies that the attacker won't be able to remove the watermark with slight distortion. Furthermore, a larger block size s helps our watermarking methodology be more robust.

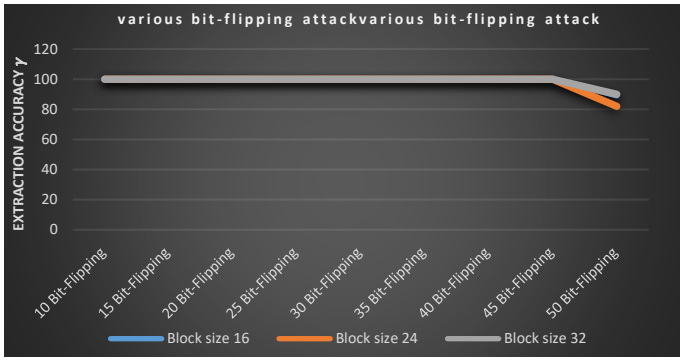

Fig. 7: Extraction accuracy γ under various bit-flipping attack.

VI. Conclusion

This paper implements and proposes a new SCBIR scheme in the context of cloud computing. Each image is represented by a single compact aggregated vector derived from local descriptors. The computation and communication costs are significantly reduced using this method. CS can calculate the resemblance scores for the encrypted image feature vectors without decryption or additional communication because the aggregated vectors are encrypted using the ASPE algorithm. To improve search efficiency, the image feature vectors are indexed as tree-index. from $O(n)$ to $O(n')$. A variable number of visual words are used to create the aggregated vectors, ORB and SIFT is the two popular local descriptors used in our research. The proposed scheme also takes into account dishonest DU in our schemes so propose a watermark-based procedure to prevent illegal images distribution. Generally speaking, image features are safe and protected even against Ciphertext-only Attack scenario as proven in [12], image components are safe and protected against the Chosen-plaintext Attack scenario as proven in [16, 23, 24, 26], In the future research will try improving invisible watermarks to prevent dishonest DU from distributing images illegally. Our proposed scheme's practical value is demonstrated by the results.

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," in *Storage and retrieval for still image and video databases iv*, 1996, vol. 2670: International Society for Optics and Photonics, pp. 426-437.

[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] C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, "Secure and robust SIFT," in *Proceedings of the 17th ACM international conference on Multimedia*, 2009, pp. 637-640.

[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," in *2011 International conference on computer vision*, 2011: Ieee, pp. 2564-2571.

[9] 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.

[10] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE computer society conference on computer vision and pattern recognition*, 2010: IEEE, pp. 3304-3311.

[11] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169-178.

[12] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 139-152.

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

[14] L. Zhang *et al.*, "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.

[15] 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.

[16] J. Qin *et al.*, "An encrypted image retrieval method based on Harris corner optimization and LSH in cloud computing," *IEEE Access*, vol. 7, pp. 24626-24633, 2019.

[17] 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.

[18] 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.
[19] 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.
[20] 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.
[21] 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.
[22] B. Ferreira, J. Rodrigues, J. Leitao, and H. Domingos, "Privacy-preserving content-based image retrieval in the cloud," in *2015 IEEE 34th symposium on reliable distributed systems (SRDS)*, 2015: IEEE, pp. 11-20.
[23] 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.
[24] 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.
[25] 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.
[26] 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.
[27] J. Anju and R. Shreelekshmi, "Secure content-based image retrieval using combined features in cloud," in *International Conference On Distributed Computing And Internet Technology*, 2020: Springer, pp. 179-197.
[28] H. Cui, X. Yuan, and C. Wang, "Harnessing encrypted data in cloud for secure and efficient mobile image sharing," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1315-1329, 2016.
[29] M. Douze, A. Ramisa, and C. Schmid, "Combining attributes and fisher vectors for efficient image retrieval," in *CVPR 2011*, 2011: IEEE, pp. 745-752.
[30] Q. Gu, Z. Xia, and X. Sun, "MSPPIR: Multi-source privacy-preserving image retrieval in cloud computing," *arXiv preprint arXiv:2007.12416*, 2020.
[31] C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, "Image feature extraction in encrypted domain with privacy-preserving SIFT," *IEEE transactions on image processing*, vol. 21, no. 11, pp. 4593-4607, 2012.
[32] A. I. Abdul-Sada, "Hiding data using LSB-3," *J. Basrah Researches (Sciences)*, vol. 33, no. 4, pp. 81-88, 2007.
[33] Z. Qin, J. Yan, K. Ren, C. W. Chen, and C. Wang, "Towards efficient privacy-preserving image feature extraction in cloud computing," in *Proceedings of the 22nd ACM international conference on multimedia*, 2014, pp. 497-506.
[34] 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.
[35] J. Anju and R. Shreelekshmi, "A faster secure content-based image retrieval using clustering for cloud," *Expert Systems with Applications*, vol. 189, p. 116070, 2022.
[36] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Computer Vision, IEEE International Conference on*, 2003, vol. 3: IEEE Computer Society, pp. 1470-1470.
[37] A. Piva, F. Bartolini, and M. Barni, "Managing copyright in open networks," *IEEE Internet Computing*, vol. 6, no. 3, pp. 18-26, 2002.
[38] N. Memon and P. W. Wong, "A buyer-seller watermarking protocol," *IEEE Transactions on image processing*, vol. 10, no. 4, pp. 643-649, 2001.
[39] M. Deng, T. Bianchi, A. Piva, and B. Preneel, "An efficient buyer-seller watermarking protocol based on composite signal representation," in *Proceedings of the 11th ACM Workshop on Multimedia and Security*, 2009, pp. 9-18.
[40] A. Rial, M. Deng, T. Bianchi, A. Piva, and B. Preneel, "A provably secure anonymous buyer–seller watermarking protocol," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 920-931, 2010.
[41] X. Zhang, "Reversible data hiding in encrypted image," *IEEE signal processing letters*, vol. 18, no. 4, pp. 255-258, 2011.