Information Sciences 387 (2017) 195–204

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# EPCBIR: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing

Zhihua Xia[a], Neal N. Xiong[b,*], Athanasios V. Vasilakos[c], Xingming Sun[a]

[a] Jiangsu Engineering Center of Network Monitoring, Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, College of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[b] Dept. of Business and Computer Science, Southwestern Oklahoma State University, Weatherford, OK, 73096, USA
[c] Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, SE-931 87 Skellefteå, Sweden

ARTICLE INFO

*Article history:*
Received 30 December 2015
Revised 15 December 2016
Accepted 24 December 2016
Available online 24 December 2016

*Keywords:*
Searchable encryption
Content-based image retrieval
Secure k-nearest neighbors algorithm
Locality-sensitive hashing

ABSTRACT

The content-based image retrieval (CBIR) has been widely studied along with the increasing importance of images in our daily life. Compared with the text documents, images consume much more storage and thus are very suitable to be stored on the cloud servers. The outsourcing of CBIR to the cloud servers can be a very typical service in cloud computing. For the privacy-preserving purposes, sensitive images, such as medical and personal images, need to be encrypted before being outsourced, which will cause the CBIR technologies in plaintext domain unusable. In this paper, we propose a scheme that supports CBIR over the encrypted images without revealing the sensitive information to the cloud server. Firstly, the feature vectors are extracted to represent the corresponding images. Then, the pre-filter tables are constructed with the locality-sensitive hashing to increase the search efficiency. Next, the feature vectors are protected by the secure k-nearest neighbor (kNN) algorithm. The security analysis and experiments show the security and efficiency of the proposed scheme.



## 1. Introduction

With the development of the imaging devices, such as digital cameras, smartphones and medical imaging equipments, our world has been witnessing a tremendous growth in quantity, availability and importance of images. The needs of efficient image storage and retrieval services are reinforced by the increase of large-scale image databases among all kinds of areas. Meanwhile, after the development of more than twenty years, CBIR techniques show maturity to be helpful in many real-word image retrieval applications [2,12,44–46]. For example, clinicians can use CBIR to find the similar cases of patients and facilitate the clinical decision-making processes. However, a large image database usually consists of millions of images. Sometimes, one digital image might be of 20 million dimensions and its size could be more than 40 megabytes, such as mammography images [15]. Therefore, the CBIR service typically incurs high storage and computation complexities. Cloud computing offers a great opportunity for the on-demand access to the ample computation and storage resources, which makes it an attractive choice for the image storage and CBIR outsourcing [41]. By outsourcing CBIR services to the cloud

* Corresponding author.
*E-mail addresses:* xia_zhihua@163.com (Z. Xia), xiongnaixue@gmail.com, neal.xiong@swosu.edu (N.N. Xiong), vasilako@ath.forthnet.gr (A.V. Vasilakos), sunnudt@163.com (X. Sun).

http://dx.doi.org/10.1016/j.ins.2016.12.030
0020-0255/

server, the data owner no longer needs to maintain the image database locally, and the authorized image users can query the cloud server for the CBIR service without interacting with the image owner.

Despite the tremendous benefits, the image privacy becomes the biggest concern about the CBIR outsourcing [1,17,18,27,33,37]. For example, the patients may not want to disclose their medical images to any others except a specific doctor in medical CBIR applications. This paper discusses the problems of privacy-preserving CBIR outsourcing with an honest-but-curious cloud server. The main contributions are summarized as follows:

1. The secure kNN algorithm is employed to protect the feature vectors, which enables the cloud server to rank the search results very efficiently without the additional communication burdens.
2. The pre-filter tables are constructed by locality-sensitive hashing to cluster the similar images. An index of two layers is constructed. The upper one consists of the pre-filter tables, which helps to increase the search efficiency. The lower one is the one-one map index which can be used to rank the search results.
3. Two typical visual descriptors, which are defined in MPEG-7, are employed to test the efficient of the scheme. The proposed methods can be flexibly generalized to the CBIR methods which are based on Euclidian distance of the feature vectors.

The rest of this paper is organized as follows. Section 2 introduces the related works. Section 3 gives a brief introduction to the system and threat models, design goals, and preliminaries. The proposed scheme is described in Section 4. The security of the scheme is analyzed in Section 5. The performance evaluations are presented in Section 6. Section 7 gives the conclusions.

## 2. Related works

Searchable encryption (SE) schemes enable the query user to search the encrypted data collections. Most of the existing SE schemes focus on the retrieval of text documents. Some early basic schemes explore the Boolean search to identify whether or not a query term is present in an encrypted text document [6,29]. Afterward, a plenty of methods are proposed under different threat models to achieve various search functionalities, such as similarity search [30,38], multi-keyword ranked search [9,10,16,40,44], dynamic search [14,36], etc. However, few of these schemes are straightforwardly appropriate to an image retrieval task. Shashank et al. [28] proposes a private content-based image retrieval (PCBIR) scheme which protects the privacy of the query image, but exposing the unencrypted image database to the server directly. Some researchers outsource the computation of image feature extraction to the could server in a privacy-preserving manner [25,32], which can be the key techniques to the privacy-preserving CBIR outsourcing. Nevertheless, the index construction and similar search on the encrypted features need to be further addressed. In addition, the homomorphic-encryption based schemes usually incur high complexities of time and storage [13,42,43].

To the best of our knowledge, Lu et al. [19] construct the first privacy-preserving CBIR scheme over the encrypted images. The authors extract the visual words to represent the images, and then calculate the Jaccard similarity between two sets of visual words so as to evaluate the similarity between the two corresponding images. The order-preserving encryption and min-hash algorithm are employed to protect the information of visual words. In another work, Lu et al. [20] investigate three image feature protection techniques, i.e. the bitplane randomization, random projection and randomized unary encoding. The features encrypted with the bitplane randomization and randomized unary encoding can be used to calculate the Hamming distance in the encryption domain. The features encrypted with the random projection can be used to calculate the L1 distance in the encryption domain. Similar to [20], Cheng et al. [3] design a CBIR system by utilizing the bitplane randomization and random projection. In 2014, Lu et al. [21] compared their three feature protection methods with the homomorphic encryption. The author indicated that the homomorphic encryption is too time-consuming to construct a practical SE scheme. Ferreira *et al.* [8] propose an image encryption method which is suitable for the privacy-preserving CBIR. In [8], the color information is separated from the texture information. The texture information is encrypted by the probabilistic cryptosystem to protect the image content, but the color information is encrypted by the deterministic cryptosystem to support the color-feature based CBIR. Cheng et al. [5] propose a markov-process based retrieval scheme for the encrypted images. The images are encrypted by the stream cipher, and the markov features are directly extracted from the encrypted image data. Support vector machine [11,35] is utilized to classify the images. Xia et al. [39] propose a privacy-preserving CBIR scheme using the local features and earth mover's distance (EMD). A linear transformation is applied to protect the sensitive information in the calculation of EMD.

## 3. Problem formulation

### 3.1. System and treat models

The system model in this paper involves three different types of entities: the image owner, image user and cloud server, as illustrated in Fig. 1.

**Image owner** wants to outsource his local data, i.e. a collection of $n$ images $\mathcal{M} = \{m_1, m_2, \ldots, m_n\}$, to the cloud server in encrypted form $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$, meanwhile keeping the capability to search over the encrypted images. Firstly, the

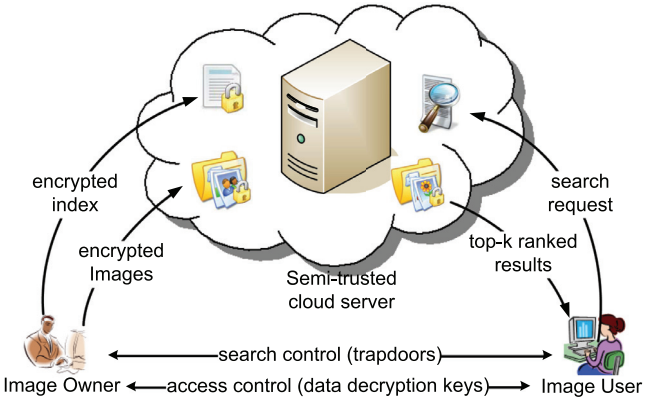


**Fig. 1.** The framework of privacy-preserving CBIR scheme.

image owner extracts feature vectors $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ from $\mathcal{M}$, and then constructs a secure searchable index $\mathcal{I}$ on $\mathcal{F}$. Next, both the encrypted image collection $\mathcal{C}$ and the index $\mathcal{I}$ are outsourced to the cloud server. The image owner also takes the responsibility to authorize image users through a certain secure method, which is orthogonal to the SSE schemes and will not be discussed in this paper as many previous SSE schemes [6,8,29,36].

**Image users** are the authorized ones to retrieve images from the cloud server. To request a search, the image user firstly generates a trapdoor *TD* for the query image, and then submits the trapdoor *TD* to the cloud server. After receiving the resulting images, the user can decrypt them with the secret key shared by the image owner.

**Cloud server** stores the encrypted image collection $\mathcal{C}$ and the index $\mathcal{I}$ for the image owner and processes the query requests from the users. We consider that the cloud server is "honest-but-curious", which means the cloud server will correctly follow the protocol specification, but will keep and analyze the communication history so as to obtain the sensitive information. Thus, the privacies of the image content, the image features and the trapdoors need to be properly protected.

Just as the previous SE schemes [6,8,29,36], it is easy to deduce that the images $m_i$ and $m_j$ are similar to each other if both the images $m_i$ and $m_j$ have the high similarity scores to the same query image. This type of information leakage is not considered in this paper.

### 3.2. Design goals

**Efficiency.** The linear search is quite inefficient and computationally impracticable for a large database. The proposed scheme aims to achieve a better-than-linear search efficiency through constructing an efficient index.

**Security.** The plaintext data regarding the image content, image features and trapdoors needs to be kept unknown to the cloud server.

### 3.3. Preliminaries

**MPEG-7 visual descriptors.** MPEG-7 is a multimedia content description standard which offers a comprehensive set of descriptors for the multimedia data description [22]. In this paper, two MPEG-7 descriptors are utilized:

1. The color layout descriptor (CLD) provides information about the spatial color distribution within images. After an image is divided into 64 blocks, CLD descriptor is extracted from each of these blocks based on the discrete cosine transform.
2. The edge histogram descriptor (EHD) captures the spatial distribution of edges. The distribution of edges is a good texture signature for the image matching even when the underlying texture is not homogeneous.

Both the two descriptors can be represented as the feature vectors, and the image similarity could be measured by Euclidean distance between the feature vectors. For more specific descriptions about these descriptors, please refer to [22].

**Locality-sensitive hashing.** Locality-sensitive hashing (LSH) has the property that the close items will collide with a higher probability than the distant ones, which can be applied in approximate queries [7]. A hash function family $\mathcal{H} = \{h : \mathcal{S} \to \mathcal{U}\}$ is called $(c, cr, p_1, p_2)$-sensitive for any $x, y \in \mathcal{S}$ if

$$\begin{cases} Pr\{h(x) = h(y)\} \geq p_1 & for \quad d(x,y) \leq d_0 \\ Pr\{h(x) = h(y)\} \leq p_2 & for \quad d(x,y) \geq cd_0, \end{cases} \tag{1}$$

where the constant $c > 1$ and probabilities $p_1 > p_2$.

To enlarge the gap between $p_1$ and $p_2$, multiple hash functions can be jointed to construct a new function family $\mathcal{G} = \{g : \mathcal{S} \to \mathcal{U}^{\lambda}\}$ where $g(v) = (h_1(v), h_2(v), \ldots h_{\lambda}(v))$, $h_i \in \mathcal{H}$ is the concatenation of $\lambda$ LSH functions. In practice, multiple hash tables can be constructed with multiple $g_i \in \mathcal{G}$. The set of results is a union from these multiple hash tables.

**Image owner side:**

- $\mathcal{K} \leftarrow \mathsf{KeyGen}(1^{\kappa})$ is the key generation algorithm that takes the security parameter $\kappa$ as the input, and returns the secret key set $\mathcal{K}$.
- $\mathcal{I} \leftarrow \mathsf{IndexGen}(\mathcal{K}, \mathcal{M})$ is the index generation algorithm that takes the secret key set $\mathcal{K}$ and the image collection $\mathcal{M}$ as the inputs, and returns the index $\mathcal{I}$.
- $\mathcal{C} \leftarrow \mathsf{ImgEnc}(\mathcal{K}, \mathcal{M})$ is the image encryption algorithm that takes the secret key set $\mathcal{K}$ and the image collection $\mathcal{M}$ as the inputs, and returns the encrypted image collection $\mathcal{C}$.

**Image User side:**

- $TD \leftarrow \mathsf{TrapdoorGen}(\mathcal{K}, m_q)$ is the trapdoor generation algorithm that takes $\mathcal{K}$ and the query image $m_q$ as the inputs, and returns the query trapdoor $TD$.
- $\mathcal{M}_q \leftarrow \mathsf{ImgDec}(\mathcal{K}, \mathcal{R}_q)$ is the decryption algorithm that takes $\mathcal{K}$ and a retrieved set of encrypted images $\mathcal{R}_q$ as the inputs, and returns the set of decrypted images $\mathcal{M}_q$.

**Cloud server side:**

- $\mathcal{R}_q \leftarrow \mathsf{Search}(\mathcal{I}, \mathcal{C}, TD)$ is the search algorithm that takes the encrypted collection $\mathcal{C}$, the index $\mathcal{I}$ and the trapdoor $TD$ as the inputs, and returns the search result set $\mathcal{R}_q$.

**Fig. 2.** Overview of the algorithms in the proposed scheme.

In our scheme, LSH based on the $p$-stable distribution is utilized to construct the pre-filter tables. A $p$-stable LSH $h_{a,\,b}$: $R^l \rightarrow Z$ maps an $l$-dimensional vector $v$ into an integer [7], and can be formulated as $h_{a,b}(v) = \lfloor \frac{(a \cdot v + b)}{r} \rfloor$, where $a$ is an $l$-dimensional random vector with the entries following a $p$-stable distribution, $b$ is a real number chosen uniformly from the range $[0, r)$, and $r$ is an integer constant.

## 4. The proposed scheme

### 4.1. Overview of the proposed scheme

The proposed scheme consists of a tuple of algorithms which are executed by different entities. A brief description of the algorithms is presented in Fig. 2.

Given an image collection $\mathcal{M}$, the image owner runs KeyGen, IndexGen and ImgEnc to generate the set of security keys $\mathcal{K}$, the secure index $\mathcal{I}$ and the encrypted image collection $\mathcal{C}$, respectively. Next, the image owner outsources the index $\mathcal{I}$ and the collection $\mathcal{C}$ to the cloud server, and then sends the key set $\mathcal{K}$ to the authorized image users.

In order to retrieve the similar images, the authorized image user runs TrapdoorGen to generate a query trapdoor $TD$, and then submits the trapdoor $TD$ and his user identity UID to the cloud server. Upon receiving the search request, the cloud server runs Search to obtain a temporary result set $\mathcal{R}_q$ including the top-$k$ most similar images.

### 4.2. Specifications of the algorithms

In this subsection, we will present the details of our privacy-preserving CBIR scheme, including KeyGen and IndexGen on the image owner side, TrapdoorGen on the image user side, and Search on the cloud server side. Please note that the image files on the cloud server can be encrypted with a standard encryption algorithm such as Advanced Encryption System (AES). Thus, the encryption of the images is not discussed in the paper.

- $\mathcal{K} \leftarrow \mathsf{KeyGen}(1^{\kappa})$ is the key generation algorithm that takes the security parameter $\kappa$ as the input, and returns the set of security keys $\mathcal{K} = \{\mathbf{S}, \mathbf{M}_1, \mathbf{M}_2, \{g_j\}_{j=1}^{L}, \{k_j\}_{j=1}^{L}, k_{img}\}$. Here, $\mathbf{S}$ is a vector of $l+1$ bits, $\mathbf{M}_1$ and $\mathbf{M}_2$ are two invertible matrices with the size $(l+1) \times (l+1)$ , $\{g_j\}_{j=1}^{L}$ is the set of LSH functions, $\{k_j\}_{j=1}^{L}$ is the set of security keys for the bucket encryption, and $k_{img}$ is the secret key for the image encryption.
- $\mathcal{C} \leftarrow \mathsf{ImgEnc}(\mathcal{K}, \mathcal{M})$ is the image encryption algorithm that takes the secret key set $\mathcal{K}$ and the image collection $\mathcal{M}$ as the inputs, and returns the encrypted image collection $\mathcal{C}$. Digital images have some characteristics such as unfair distribution of pixel values and strong correlation between neighboring pixels. As a result, the traditional encryption methods are not quite suitable to images for security and efficiency issues. In this paper, we employ chaotic maps to protect the image content [24].

**Table 1**
One-one map index.

| Image identity | Feature vector |
|---|---|
| $\mathrm{ID}(m_1)$ | $f_1$ |
| $\mathrm{ID}(m_2)$ | $f_2$ |
| ... | ... |
| $\mathrm{ID}(m_i)$ | $f_i$ |
| ... | ... |
| $\mathrm{ID}(m_n)$ | $f_n$ |

**Table 2**
The $j$th pre-filter table.

| Table key | Image identities |
|---|---|
| $BKT_{j,1}$ | $\mathrm{ID}(m_1), \mathrm{ID}(m_{25}), \mathrm{ID}(m_{49}), \mathrm{ID}(m_{113})$ |
| $BKT_{j,2}$ | $\mathrm{ID}(m_{14}), \mathrm{ID}(m_{56}), \mathrm{ID}(m_{104}), \mathrm{ID}(m_{217})$ |
| ... | ... |
| $BKT_{j,N_j}$ | $\mathrm{ID}(m_{32}), \mathrm{ID}(m_{73}), \mathrm{ID}(m_{120}), \mathrm{ID}(m_{315})$ |

- $\mathcal{I} \leftarrow \mathsf{IndexGen}(\mathcal{K}, \mathcal{M})$ is the index generation algorithm that takes the secret key set $\mathcal{K}$ and the image collection $\mathcal{M}$ as the inputs, and returns the index $\mathcal{I}$. For clarification, we divide the process of index generation into two steps: the generation of unencrypted index and the index encryption.

  **Step1: the generation of unencrypted index.** At the beginning, a feature vector $f_i = (f_{i,1}, f_{i,2}, \ldots, f_{i,l})^T$ is extracted from each image $m_i \in \mathcal{M}$ using the feature extraction methods described in Section 3.3. Then, the similarity between two images depends on the similarity between the two corresponding feature vectors. Intuitively, a one-one map index can be constructed for the retrieval purpose. However, this will cause a linear time complexity. In order to get better search efficiency, a two-layer index is constructed in the proposed scheme. Specifically, the bottom layer is the one-one map index which is illustrated in Table 1. The upper one consists of the pre-filter tables which are constructed on basis of the one-one map index. During the search process, most of the dissimilar images are discarded according to the pre-filter tables. Next, the similarity scores of the remaining images to the query image are calculated and ranked according to the one-one map index.

  The construction of the one-one map index is quite straightforward, and the constructions of the pre-filter tables are completed by the LSH which is introduced in Section 3.3. Specifically, the image owner randomly chooses $\lambda$ LSH functions $h_1, h_2, \ldots, h_\lambda \in \mathcal{H}$ and applies $g(f_i) = (h_1(f_i), h_2(f_i), \ldots, h_\lambda(f_i))$ to all the features in $\{f_i\}_{i=1}^n$ so as to build a pre-filter table. As introduced in Section 3.3, the function $g(\cdot)$ maps an $l$-dimensional vector into $\lambda$ integers, which form a $\lambda$-dimensional vector called bucket. The images with the same bucket value can be treated as a cluster of similar images. To provide more possible results, this process is repeated $L$ times to generate $L$ pre-filter tables. To sum up, the set of buckets can be denoted as $\{BKT_{j,b}\}_{j\in[1,L],b\in[1,N_j]}$, where $N_j$ refers to the total number of buckets in the $j$th pre-filter table. In the proposed scheme, each image $m_i \in \mathcal{M}$ is mapped into $L$ buckets. An example of the pre-filter table is illustrated in Table 2.

  **Setp2: the index encryption.** The image feature vectors in plaintext may reveal information about the image content [3,4,8,19,20,39]. For example, a color histogram with large blue component would indicate the likely presence of the sky or ocean, and the shape descriptors may disclose the information about the likely object in the image. Therefore, the feature vectors in the one-one map index need to be encrypted. The key point is to maintain that the encrypted feature vectors can be still used to calculate and rank the similarities. Intuitively, the homomorphic encryption techniques can be employed here. However, the homomorphic encryption is generally time-consuming and leads an additional round of communication between the cloud server and the query user [26]. As an alternative, the secure kNN algorithm [34] is employed to protect the feature vectors $\{f_i\}_{i=1}^n$. Specifically, for a feature vector $f_i = (f_{i,1}, f_{i,2}, \ldots, f_{i,l})^T$, we modify it into $\widehat{f}_i = (f_{i,1}, f_{i,2}, \ldots, f_{i,l}, \|f_i\|^2)^T$, where $\|f_i\|$ is the Euclidean norm of $f_i$. Next, we split $\widehat{f}_i$ into two random vectors $\{\widehat{f}_{ia}, \widehat{f}_{ib}\}$ according to $\mathbf{S}$ as: if $\mathbf{S}[j] = 0$, $\widehat{f}_{ia}[j]$ and $\widehat{f}_{ib}[j]$ are set equal to $\widehat{f}_i[j]$; if $\mathbf{S}[j] = 1$, $\widehat{f}_{ia}[j]$ and $\widehat{f}_{ib}[j]$ are set as two random values whose sum equals to $\widehat{f}_i[j]$. Finally, the encrypted feature vector is generated as $f_i' = \{\mathbf{M}_1^T\widehat{f}_{ia}, \mathbf{M}_2^T\widehat{f}_{ib}\}$.

  In addition, LSH is not necessary to have the one-way property. Thus, we cannot directly outsource the pre-filter tables to the cloud server as the bucket values may disclose the information about the features. To enhance the security, the bucket values are protected by a one-way hash function $\phi$, as illustrated in Table 3. Finally, the encrypted index $\mathcal{I}$, including the one-one map index and the pre-filter tables, is uploaded to the cloud server for the secure CBIR.
- $TD \leftarrow \mathsf{TrapdoorGen}(\mathbf{S}, \mathbf{M}_1, \mathbf{M}_2, \{g_j\}_{j=1}^L, \{k_j\}_{j=1}^L, m_q)$. To retrieve similar images, a query user generates a trapdoor and sends it to the cloud server. The trapdoor cannot reveal the information about the query image but can be used to search similar images on index $\mathcal{I}$. The procedure of $\mathsf{TrapdoorGen}$ is defined as follows:

  1. Calculate the feature vector $f_q$ from the query image $m_q$.

**Table 3**
The $j$th pre-filter table with the bucket value encrypted.

| Table key | Image identities |
|---|---|
| $\phi(BKT_{j,1}, k_j)$ | ID($m_1$), ID($m_{25}$), ID($m_{49}$), ID($m_{113}$) |
| $\phi(BKT_{j,2}, k_j)$ | ID($m_{14}$), ID($m_{56}$), ID($m_{104}$), ID($m_{217}$) |
| ... | ... |
| $\phi(BKT_{j,N_j}, k_j)$ | ID($m_{32}$), ID($m_{73}$), ID($m_{120}$), ID($m_{315}$) |

2. For each $j \in [1, L]$, compute the bucket value $BKT_j = g_j(f_q)$, and then encrypt $BKT_j$ to be $\phi(BKT_j, k_j)$ using the secret key $k_j$.
3. Modify query vector $f_q = (f_{q,1}, f_{q,2}, \ldots, f_{q,l})^T$ into $\widehat{f}_q = (-2f_{q,1}, -2f_{q,2}, \ldots, -2f_{q,l}, 1)^T$, and then split $\widehat{f}_q$ into two random vectors $\{\widehat{f}_{qa}, \widehat{f}_{qb}\}$ as: if $\mathbf{S}[j] = 0$, $\widehat{f}_{qa}[j]$ and $\widehat{f}_{qb}[j]$ are set as two random values whose sum equals to $\widehat{f}_q[j]$; if $\mathbf{S}[j] = 1$, $\widehat{f}_{qa}[j]$ and $\widehat{f}_{qb}[j]$ are set equal to $\widehat{f}_q[j]$. Note that the split operation here is a little different from that in IndexGen. Then, the query vector is encrypted as $f'_q = \{\gamma \mathbf{M}_1^{-1}\widehat{f}_{qa}, \gamma \mathbf{M}_2^{-1}\widehat{f}_{qb}\}$, where $\gamma \in R$ is a random positive value.
4. Finally, the trapdoor is generated as $TD = \{\{\phi(BKT_j, k_j)\}_{j=1}^{L}, f'_q\}$.

- $\mathcal{R} \leftarrow \text{Search}(\mathcal{I}, \mathcal{C}, TD)$. Upon receiving a trapdoor $TD$ from an image user, the cloud server executes Search to search the similar images.
  1. Firstly, the cloud server fetches the image IDs from $L$ pre-filter tables according to the encrypted bucket values $\{\phi(BKT_j, k_j)\}_{j=1}^{L} \in TD$. With the property of locality-sensitive hashing, these images are likely to be similar to the query image. This step removes the dissimilar images very efficiently.
  2. Next, the distances of the images fetched above to the query image are calculated and ranked, which helps to reduce the communication burden by just sending the top-$k$ most similar images to the query user. The distance between a database feature vector $f_i$ and a query feature vector $f_q$ is calculated as follows,

$$\begin{aligned} f_q'^T f'_i &= (\gamma M_1^{-1}\widehat{f}_{qa})^T M_1^T \widehat{f}_{ia} + (\gamma M_2^{-1}\widehat{f}_{qb})^T M_2^T \widehat{f}_{ib} \\ &= \gamma(\widehat{f}_{qa})^T \widehat{f}_{ia} + \gamma(\widehat{f}_{qb})^T \widehat{f}_{ib} \\ &= \gamma(\widehat{f}_q)^T \widehat{f}_i \\ &= \gamma(\|f_i\|^2 - 2\sum_{j=1}^{l} f_{i,j} f_{q,j}) \\ &= \gamma(\|f_q - f_i\|^2 - \|f_q\|^2). \end{aligned} \tag{2}$$

  The distance $\|f_q - f_i\|^2$ is hidden by the secret scalar $r$ and the unknown $\|f_q\|^2$. And $f_q'^T f'_1 > f_q'^T f'_2$ implies $\|f_q - f_1\|^2 > \|f_q - f_2\|^2$, which means that the cloud server can directly find closest feature vectors by simply sorting the set of vector products $f_q'^T f'_i$, without knowing the original feature vectors.
  3. Finally, the cloud server puts the top-$k$ most similar encrypted images into the set $\mathcal{R}_q$ which will be sent to the query user.

## 5. Security analysis

Similar to the previous SE scheme [3,4,8,19,20,39], the cloud server is considered to be "honest-but-curious", which means that the cloud server will correctly follow the protocol specification but keep curious to analyze the information about the images. Thus, the privacies of the image features and the trapdoors need to be properly protected.

1. **The privacy of the image.** In our paper, the images are encrypted by the chaotic maps. Chaotic maps have many fundamental properties which can be considered analogous to the properties of ideal ciphers such as confusion and diffusion. The chaotic maps can protect the histogram and correlation statistics well [24]. In Fig. 3, we illustrate an image and its corresponding encrypted version by chaotic map.
2. **The privacy of the image features.** The image features may reveal the information about image content. In our scheme, the feature vectors are encrypted by the secure kNN algorithm which is proved to be secure against the Chipertext-only Attack (COA) model [34]. In addition, the bucket values that are mapped from the feature vectors are further protected by a one-way hash function. Thus, the information about the feature vectors will not be leaked from these bucket values.
3. **The privacy of the trapdoors.** Similar to the feature vectors in the index, the feature vectors in trapdoors are encrypted by the secure kNN algorithm. And the bucket values in the trapdoors are also further protected to prevent the information leakage. Thus, the privacy of the trapdoors is also well protected.
4. **The leakage of the similarity information.** In our scheme, the information about the similarity among images are leaked. For instance, if the images $m_i$ and $m_j$ are returned as the search results to the same query, it is easy to deduce that the images $m_i$ and $m_j$ are similar to each other. In addition, we use the pre-filter tables to group the similar

(a) (b)

**Fig. 3.** Lena image and its encrypted version by chaotic map.

**Table 4**
Parameters in the experiments.

| Visual descriptors | Parameters | | | |
|---|---|---|---|---|
| | $l$ | $L$ | $\lambda$ | $r$ |
| CLD | 120 | 4 | 2 | 4 |
| EHD | 80 | 2 | 2 | 4 |

images to improve the search efficiency. Thus, the cloud server knows those images in the same bucket are similar to each other. This type of information leakage is a compromise for efficiency.

## 6. Performance evaluation

In this section, we present the performance evaluations of the proposed scheme on Corel image dataset which is a commonly used dataset for the image retrieval performance test [31]. This image database includes 100 categories of images and each category contains 100 similar images. The entire secure search scheme is implemented using C++ language on a Windows 7 operation system with Intel Core(TM) Duo Processor 2.80 GHz. The performance of the scheme depends on several parameters including the parameter in the hash function $r$, the number of jointed LSH functions in the construction of pre-filter table $\lambda$, the number of the pre-filter tables $L$, and the dimensionality of the visual descriptor $l$. In this paper, these parameters are set experimentally except $l$ which is fixed in a specific feature extraction algorithm. The parameters in our experiments are summarized in Table 4. Please note that the parameters used here are not claimed to be the optimum ones.

### 6.1. Retrieval precision

In our experiments, the "precision" for a query is defined as that in [23]: $P_k = k'/k$, where $k'$ is the number of real similar images in the $k$ retrieved images. According to the Eq. 2, the encryption of feature vector will not influence the retrieval precision. However, the pre-filter tables which are utilized to improve the search efficiency will affect the retrieval precision.

Two MPEG-7 descriptors are adopted to test the retrieval precisions. The retrieval precisions of our schemes with and without the pre-filter tables are compared. We choose 20 image categories to test the retrieval precision. Two samples from each category are randomly selected as the query images. Accordingly, the retrieval precision are averaged from 40 queries for each visual descriptor.

The average precisions of the two descriptors are presented in Fig. 4. The precisions are mainly dependent on the performances of the visual descriptors. The usage of the pre-filter tables decreases the retrieval precision. The average decline rates are 19.90% and 13.93% for CLD and EHD, respectively. These losses of precision are traded off for the search efficiency.

### 6.2. Efficiency

The time consumptions of the index construction, trapdoor generation and search operation are tested in this subsection. In addition, the storage consumption of the index is presented.

#### 6.2.1. Time consumption of the index construction

Before the index construction, we have completed the extraction of the feature vectors. Thus, the time consumption of the index construction in our experiments mainly includes two parts: 1) the consumption for building $L$ hash tables, and

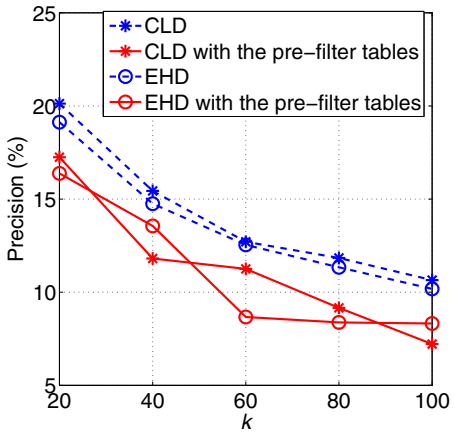


**Fig. 4.** Average retrieval precision when top-$k$ results are retrieved.

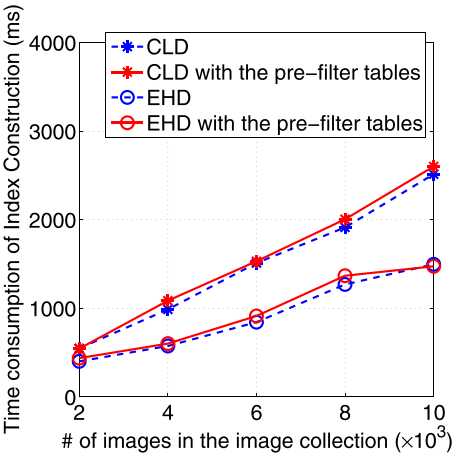


**Fig. 5.** Time consumption of the index construction.

**Table 5**
Time consumed by the trapdoor generation (ms).

| Descriptor | with the pre-filter tables | without the pre-filter tables |
|---|---|---|
| CLD | 3.476 | 3.558 |
| EHD | 1.252 | 1.27 |

2) the consumption for encrypting the feature vectors with a splitting operation and two multiplicative operations with the $(l+1)\times(l+1)$ matrices. In order to construct a pre-filter table, it takes $O(n\lambda l)$ time to generate the bucket values, where $n$ denotes the total number of the images, $l$ denotes the dimensionality of the feature vector, and $\lambda$ denotes the number of the jointed hash functions. The time complexity of the splitting operation is $O(nl)$ and the time complexity of the matrix multiplication is $O(nl^2)$. In total, the time complexity for the index construction is $O(Ln\lambda l + nl + nl^2)$. Since $L$, $\lambda$ and $l$ are the fixed constants in our scheme, the time consumption of the index construction is almost linear to the size of image collection, i.e. $O(n)$. The time consumption of the index construction is illustrated in Fig. 5, which shows that the construction of the pre-filter tables costs very little time.

*6.2.2. Time consumption of the trapdoor generation*

Similar to the index generation, the trapdoor generation incurs the calculations of the bucket values, a splitting operation, and two multiplicative operations with the $(l+1)\times(l+1)$ matrices, thus the time complexity is $O(L\lambda l + l + l^2)$. The time cost of the trapdoor generation is mainly dependent on the dimensionality of the visual descriptor, as illustrated in Table 5.

*6.2.3. Time consumption of the search operation*

During the search process, the cloud server firstly retrieves $n'$ similar images from the pre-filter tables according to the bucket values submitted by the query user. Next, a linear search is executed over these $n'$ images to retrieve top-$k$ most

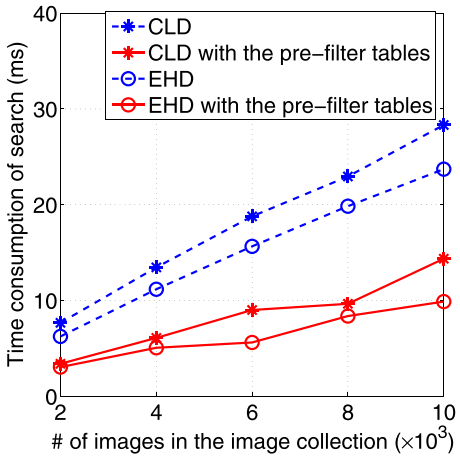


**Fig. 6.** Average search time for the different size of image collection.

**Table 6**
Storage consumption of indexes from 10,000 images.

| Index | Storage consumption (KB) | |
|---|---|---|
| | CLD | EHD |
| Without the pre-filter tables | 33,195 | 22,258 |
| With the pre-filter tables | 34,181 | 23,244 |

similar results. Thus, the search complexity of the scheme with the pre-filter tables is $O(n')$. Generally, $n'$ is expected to be far less than the database size $n$. As is illustrated in Fig. 6, the scheme with the pre-filter tables achieves better efficiency than that without the pre-filter tables. The average decline rates of time consumption are 67.12% and 58.86% for SCD and EHD, respectively. Please note that the efficiency of the scheme can be further improved with the smaller $L$, smaller $r$, and larger $\lambda$, which can be flexibly adjusted according to the real-world applications.

*6.2.4. Storage consumption of the index*

The index in the proposed scheme consists of a one-one map index and $L$ hash tables. In Table 6, we present the storage consumption of the indexes with and without the pre-filter tables. The results are calculated from 10,000 images. Table 6 shows that the pre-filter tables consume a little storage compared with the one-one map index.

## 7. Conclusions

In this paper, we presented a privacy-preserving content-based image retrieval scheme in a cloud computing scenario. The secure kNN algorithm is applied to encrypt the visual features. The similarity scores can be directly calculated and with the encrypted features by the cloud server, which enables the cloud server to rank the retrieved results without the additional communication burdens. The locality-sensitive hashing is utilized to improve the search efficiency. Overall, the image features are secure against Ciphertext-only Attack model, and the search efficiency is improved from $O(n)$ to $O(n')$. In the future, it is worth further improving the security lever of image features. The feature extraction in encrypted image are also the meaningful future works to secure CBIR outsourcing.

## Acknowledgments

This work is supported by the NSFC (61672294, 61601236, U1536206, 61502242, 61572258, U1405254, 61373133, 61373132, 61232016), BK20150925, Fund of Jiangsu Engineering Center of Network Monitoring (KJR1402), Fund of MOE Internet Innovation Platform (KJRP1403), Six peak talent project of Jiangsu Province (R2016L13), BK21+ program by the Ministry of Education of Korea, CICAEET, and PAPD fund.

## References

[1] M. Ali, S.U. Khan, A.V. Vasilakos, Security in cloud computing: opportunities and challenges, Inf. Sci. 305 (2015) 357–383.
[2] B. Chen, H. Shu, G. Coatrieux, G. Chen, X. Sun, J.L. Coatrieux, Color image analysis by quaternion-type moments, J. Math. Imaging Vis. 51 (1) (2015) 124–144.
[3] B. Cheng, L. Zhuo, Y. Bai, Y. Peng, J. Zhang, Secure index construction for privacy-preserving large-scale image retrieval, in: Proceedings of the Fourth International Conference on Big Data and Cloud Computing, IEEE, 2014, pp. 116–120.

[4] H. Cheng, X. Zhang, J. Yu, F. Li, Markov process based retrieval for encrypted jpeg images, in: Proceedings of 10th International Conference on Availability, Reliability and Security, IEEE, 2015, pp. 417–421.
[5] H. Cheng, X. Zhang, J. Yu, F. Li, Markov process-based retrieval for encrypted jpeg images, EURASIP J. Inf. Secur. 2016 (1) (2016) 1–9.
[6] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, Searchable symmetric encryption: improved definitions and efficient constructions, in: Proceedings of 13th ACM Conference on Computer and Communications Security, ACM, 2006, pp. 79–88.
[7] M. Datar, N. Immorlica, P. Indyk, V.S. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, in: Proceedings of the twentieth annual symposium on Computational geometry, ACM, 2004, pp. 253–262.
[8] B. Ferreira, J. Rodrigues, J. Leito, H. Domingos, Towards an image encryption scheme with content-based image retrieval properties, in: Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance, Springer, 2015, pp. 311–318.
[9] Z. Fu, X. Sun, Q. Liu, L. Zhou, J. Shu, Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing, IEICE Trans. Commun. E98-B (1) (2015) 190–200.
[10] Z. Fu, X. Wu, C. Guan, X. Sun, K. Ren, Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement, IEEE Trans. Inf. Forensics Secur. 11 (12) (2016) 2706–2716.
[11] B. Gu, V.S. Sheng, K.Y. Tay, W. Romano, S. Li, Incremental support vector learning for ordinal regression, IEEE Trans. Neural Netw. Learn. Syst. 26 (7) (2015) 1403–1416.
[12] K. Guo, R. Zhang, Z. Zhou, Y. Tang, L. Kuang, Combined retrieval: a convenient and precise approach for internet image retrieval, Inf. Sci. 358 (2016) 151–163.
[13] C.-Y. Hsu, C.-S. Lu, S.-C. Pei, Secure and robust sift, in: Proceedings of 17th ACM international conference on Multimedia, ACM, 2009, pp. 637–640.
[14] S. Kamara, C. Papamanthou, Parallel and dynamic searchable symmetric encryption, in: Financial Cryptography and Data Security, Springer, 2013, pp. 258–274.
[15] J.M. Lewin, R.E. Hendrick, C.J. DOrsi, P.K. Isaacs, L.J. Moss, A. Karellas, G.A. Sisney, C.C. Kuni, G.R. Cutter, Comparison of full-field digital mammography with screen-film mammography for cancer detection: results of 4,945 paired examinations 1, Radiology 218 (3) (2001) 873–880.
[16] H. Li, D. Liu, Y. Dai, T.H. Luan, X.S. Shen, Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage, IEEE Trans. Emerg. Top. Comput. 3 (1) (2015) 127–138.
[17] J. Li, X. Li, B. Yang, X. Sun, Segmentation-based image copy-move forgery detection scheme, IEEE Trans. Inf. Forensics Secur. 10 (3) (2015) 507–518.
[18] C. Liu, L. Zhu, M. Wang, Y. an Tan, Search pattern leakage in searchable encryption: attacks and new construction, Inf. Sci. 265 (2014) 176–188.
[19] W. Lu, A. Swaminathan, A.L. Varna, M. Wu, Enabling search over encrypted multimedia databases, in: Proceedings of IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, 2009. 725418–725418.
[20] W. Lu, A.L. Varna, A. Swaminathan, M. Wu, Secure image retrieval through feature protection, in: Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2009, pp. 1533–1536.
[21] W. Lu, A.L. Varna, M. Wu, Confidentiality-preserving image search: a comparative study between homomorphic encryption and distance-preserving randomization, IEEE Access 2 (1) (2014) 125–141.
[22] B.S. Manjunath, J.-R. Ohm, V.V. Vasudevan, A. Yamada, Color and texture descriptors, IEEE Trans. Circuits Syst. Video Technol. 11 (6) (2001) 703–715.
[23] H. Müller, W. Müller, D.M. Squire, S. Marchand-Maillet, T. Pun, Performance evaluation in content-based image retrieval: overview and proposals, Pattern Recognit. Lett. 22 (5) (2001) 593–601.
[24] N.K. Pareek, V. Patidar, K.K. Sud, Image encryption using chaotic logistic map, Image Vis. Comput. 24 (9) (2006) 926–934.
[25] Z. Qin, J. Yan, K. Ren, C.W. Chen, C. Wang, Towards efficient privacy-preserving image feature extraction in cloud computing, in: ACM International Conference on Multimedia, ACM, 2014, pp. 497–506.
[26] S. Rane, P.T. Boufounos, Privacy-preserving nearest neighbor methods: comparing signals without revealing them, IEEE Signal Process. Mag. 30 (2) (2013) 18–28.
[27] Y.-J. Ren, J. Shen, J. Wang, J. Han, S.-Y. Lee, Mutual verifiable provable data auditing in public cloud storage, J. Internet Technol. 16 (2) (2015) 317–323.
[28] J. Shashank, P. Kowshik, K. Srinathan, C. Jawahar, Private content based image retrieval, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[29] D.X. Song, D. Wagner, A. Perrig, Practical techniques for searches on encrypted data, in: Proceedings of IEEE Symposium on Security and Privacy, IEEE, 2000, pp. 44–55.
[30] C. Wang, K. Ren, S. Yu, K.M.R. Urs, Achieving usable and privacy-assured similarity search over outsourced cloud data, in: Proceedings of INFOCOM, IEEE, 2012, pp. 451–459.
[31] J.Z. Wang, J. Li, G. Wiederhold, Simplicity: semantics-sensitive integrated matching for picture libraries, IEEE Trans. Pattern Anal. Mach. Intell. 23 (9) (2001) 947–963.
[32] Q. Wang, J. Wang, S. Hu, Q. Zou, K. Ren, Sechog: Privacy-preserving outsourcing computation of histogram of oriented gradients in the cloud, in: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, ACM, 2016, pp. 257–268.
[33] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, A.V. Vasilakos, Security and privacy for storage and computation in cloud computing, Inf. Sci. 258 (2014) 371–386.
[34] W.K. Wong, D.W.-l. Cheung, B. Kao, N. Mamoulis, Secure knn computation on encrypted databases, in: Proceedings of 2009 ACM SIGMOD International Conference on Management of Data, ACM, 2009, pp. 139–152.
[35] Z. Xia, R. Lv, Y. Zhu, P. Ji, H. Sun, Y.-Q. Shi, Fingerprint liveness detection using gradient-based texture features, Signal Image Video Process. (2016) 1–8.
[36] Z. Xia, X. Wang, X. Sun, Q. Wang, A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data, IEEE Trans. Parallel Distrib. Syst. 27 (2) (2016) 340–352.
[37] Z. Xia, L. Zhang, D. Liu, Attribute-based access control scheme with efficient revocation in cloud computing, China Commun. 13 (7) (2016) 92–99.
[38] Z. Xia, Y. Zhu, X. Sun, L. Chen, Secure semantic expansion based search over encrypted cloud data supporting similarity ranking, J. Cloud Comput. 3 (1) (2014) 1–11.
[39] Z. Xia, Y. Zhu, X. Sun, Z. Qin, K. Ren, Towards privacy-preserving content-based image retrieval in cloud computing, IEEE Trans. Cloud Comput. PP (99) (2015). 1–1.
[40] Y. Yang, M. Ma, Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for e-health clouds, IEEE Trans. Inf. Forensics Secur. 11 (4) (2016) 746–759.
[41] J. Yin, X. Lu, H. Chen, X. Zhao, N.N. Xiong, System resource utilization analysis and prediction for cloud based applications under bursty workloads, Inf. Sci. 279 (2014) 338–357.
[42] Y. Zhang, X. Sun, B. Wang, Efficient algorithm for k-barrier coverage based on integer linear programming, China Commun. 13 (7) (2016) 16–23.
[43] P. Zheng, J. Huang, An efficient image homomorphic encryption scheme with small ciphertext expansion, in: Proc. of 21st ACM international conference on Multimedia, ACM, 2013, pp. 803–812.
[44] Y. Zheng, B. Jeon, D. Xu, Q. Wu, H. Zhang, Image segmentation by generalized hierarchical fuzzy c-means algorithm, J. Intell. Fuzzy Syst. 28 (2) (2015) 961–973.
[45] Z. Zhou, Y. Wang, J. Wu, C.N. Yang, X. Sun, Effective and efficient global context verification for image copy detection, IEEE Trans. Inf. Forensics Secur. PP (99) (2016), doi:10.1109/TIFS.2016.2601065. 1–1.
[46] Y. Zhuang, N. Jiang, Z. Wu, Q. Li, D.K. Chiu, H. Hu, Efficient and robust large medical image retrieval in mobile cloud computing environment, Inf. Sci. 263 (2014) 60–86.