Efficient Privacy-Preserving Content-Based Image Retrieval in the Cloud

Kai Huang, Ming Xu(✉), Shaojing Fu, and Dongsheng Wang

College of Computer, National University of Defense Technology, Changsha, China
{kai.huang,xuming}@nudt.edu.cn, shaojing1984@yahoo.cn,
wdsh2011@gmail.com

Abstract. Cloud storage systems are increasingly being used to host personal or organizational data of critical importance, especially for image data that needs more storage space than ordinary data. While bringing in much convenience, existing cloud storage solutions could seriously breach the privacy of users. Encryption before outsourcing images to the cloud helps to protect the privacy of the data, but it also brings challenges to perform image retrieval over encrypted data. To address this issue, considerable amount of searchable encryption schemes have been proposed in the literature. However, most existing schemes are either less secure or too computation and communication intensive to be practical. In this paper, we propose an efficient privacy-preserving content-based image retrieval scheme. We first convert the high-dimensional image descriptors to compact binary codes, and then adapt the asymmetric scalar-product-preserving encryption (ASPE) to ensure the confidentiality of the sensitive images. The security analysis and experiments show the security, accuracy and efficiency of our proposed scheme.

1 Introduction

With the popularity of digital cameras and smart phones and the development of mobile internet, which makes the acquisition and transmission of image data cheaper and faster, the amount of image data has experienced exponential growth over the past decade. For example, to improve the management of urban flows and allow for real time responses to challenges, many governments are making efforts to build the smart cities, and the most important part of which is the monitoring and surveillance system generating countless images day and night. Meanwhile, image data is usually of high resolution and therefore needs more storage space than ordinary data. Consequently, the large amount of images are far beyond the storage and processing capacity of a small company or a resource-constained organization.

Advances in cloud computing have prompted many customers to outsource their storage and computing needs. By moving their image data to the cloud, customers can avoid the costs of building and maintaining a private storage infrastructure. However, such image data as monitoring and surveillance images usually contains plenty of confidential or sensitive information. Outsourcing the

B. Cui et al. (Eds.): WAIM 2016, Part II, LNCS 9659, pp. 28–39, 2016.
DOI: 10.1007/978-3-319-39958-4_3

data to the cloud means that the data is outside its control and could potentially be granted to untrusted parties. This poses a dramatic threat to the privacy of users. To mitigate this threat, more and more users tend to encrypt their image data before outsourcing. While preserving privacy of the image data on the cloud, it brings new challenges to the image retrieval problem, that is, the problem of searching for images in the encrypted domain.

Recently, numerous schemes have been designed to implement text document search over the encrypted data, which perform either Boolean search [1,2] to find the exact match between the query strings and the index strings or similarity search [3,4] to return index strings that are similar to the query strings. Although they can be applied to keyword-based image retrieval, they are not suitable for privacy-preserving content-based image retrieval (PCBIR) which needs to calculate the distances among the encrypted high dimensional vectors. Therefore, Lu et al. [5]proposed several randomization techniques to protect the image features but approximately preserve the distance between the randomized feature vectors. Although the proposed randomization techniques are computationally efficient and can obtain high search accuracy, such transformation is shown to be not secure in practice due to the distance-preserving property. The weakness of distance-preserving encryption comes from the fact that the distance information could be recovered from the encrypted databases [6]. To enhance security, the semantically secure homomorphic encryption schemes such as Paillier and Gentry, which allow computations in the encrypted domain directly, have attracted the most attention in the literature [7–9]. Through homomorphic encryption, the similarity computation can be outsourced to the cloud without exposing the data and at the same time the search accuracy can be retained. Unfortunately, the currently established homomorphic encryption techniques are too computation and communication intensive to be practical.

To address this issue, in this paper, we propose an efficient privacy-preserving content-based image retrieval scheme which allows users to retrieve the encrypted images outsourced to the cloud server. We first adopt the BPBC (bilinear projection-based binary codes) [10] to convert the high-dimensional VLAD descriptors [11] to compact binary codes. The similarity between the images is subsequently measured through the asymmetric distance between the reduced query vector and the compact points in the database. Then we adapt the asymmetric scalar-product-preserving encryption to design PCBIR to achieve the privacy requirements in the cloud environment. We implement PCBIR over real image repositories and the experiment results show that PCBIR achieves high search accuracy with low overhead.

This paper makes the following contribution: (1) we propose an efficient construction for privacy-preserving content-based image retrieval over encrypted data; (2) we formally prove the security of our proposed scheme; (3) we experimentally show that our scheme provides increased performance and lower overhead.

The rest of this paper is organized as follows: Section 2 introduces the formalization of the problem and necessary preliminaries. In Sect. 3, we describe the detail of our proposed PCBIR. Security analysis and performance evaluation are presented in Sects. 4 and 5 respectively. We conclude the paper in Sect. 6.

2 Problem Formulation

2.1 System and Threat Model

In our scheme, we consider three main entities: the cloud server, the image data owner and the image data user, as shown in Fig. 1. Generally, images are outsourced to the cloud server by the owner, and the authorized users are allowed to access the images and retrieve the required ones. For privacy protection, the owner should first generate secret keys and encrypt the images before outsourcing. To facilitate the image retrieval, the owner should also construct the secure indexes which are submitted to the cloud along with the encrypted data. Any user who has the secret keys is permitted to access and retrieve the required image data. The secret keys can be obtained through the state-of-the-art access control policy and key distribution protocol. During the retrieval process, the authorized user first generates a secure trapdoor according to a query image. Then, he sends the trapdoor to the cloud server to perform the image retrieval over the encrypted data. Upon receiving the retrieval request from the user, the cloud server returns the required images to the user according to the similarity of the images.

Similar to previous works [12–14] found in the literature, we consider an honest-but-curious cloud environment in our threat model, which means that the cloud server honestly follows the designated protocol while curiously infers some private information of interest based on the data stored and processed on it. Meanwhile, both the authorized and unauthorized data user are semi-trusted, since they may eavesdrop on other users' image data.

2.2 Design Goals

Our goal is to design a secure and well functioning image retrieval scheme over encrypted data. Specifically, we have the following goals:

Image Similarity Retrieval. Our primary goal is to support the similarity retrieval of the images, that is to allow the users to retrieve not only the duplicate images but also the most similar images that have common objects or backgrounds.

Privacy-Preserving. The main concern in our design is to protect the privacy of the image data and prevent the cloud server or the malicious users from learning additional information of the owner, which includes the original image data, the index and the query image data.

Efficiency. Compared with existing privacy-preserving image retrieval schemes, our design should take into full account the communication and computation overhead and achieve an effective but efficient privacy-preserving image retrieval scheme.

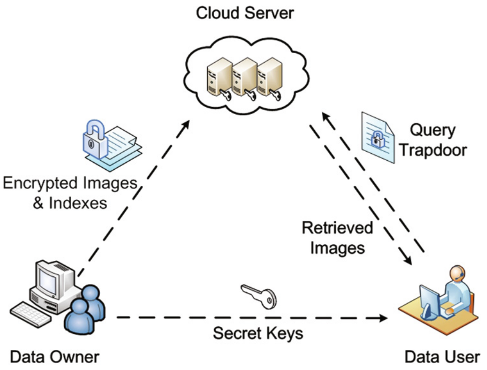

Fig. 1. System Model of Our Scheme

Adaptability. Our design should be applicable to different image representations derived from other feature extraction techniques according to the user requirements.

2.3 Preliminaries

In the following, we present some basic terminologies and algorithms that are adopted in our scheme.

Vector of Locally Aggregated Descriptors (VLAD). It is a first order extension to the popular Bag-of-Visual-Words (BoVW) model which is local features based technique for content-based image retrieval [11] and is considered to be a reliable approach to bridge the gap between low-level visual features and high-level image contents. For each image, the local features such as SIFT [15] are extracted and encoded to the corresponding visual words. Different from the BoVW model, which involves simply counting the number of descriptors associated with each cluster in the visual vocabulary and creating a histogram vector for each set of descriptors from an image, VLAD accumulates the residual of each descriptor with respect to its assigned cluster and can achieve better discrimination for the classifier.

Specifically, let $C = \{c_1, c_2, \cdots, c_k\}$ be the k clusters generated through k-means. Each d-dimensional local descriptor x from an image is associated with its nearest cluster. Then for each cluster c_i, we accumulate the difference $x - c_i$, where $c_i = NN(x)$. Representing the VLAD vector for each image by v, we have $v_{ij} = \sum_{x|x=NN(c_i)} (x_j - c_{ij})$, for $i = 1, 2, \cdots, k$ and $j = 1, 2, \cdots, d$. Often, the vector v will be power-normalized or L_2 normalized.

Then, Euclidean distance can be adopted to compute the distance of the VLAD vectors and find the k-nearest neighbor (kNN) of the query image.

Secure kNN Computation. In kNN computation, distance-preserving transformation is proved to be not secure and exact distance computation is not necessary in practice. The asymmetric scalar-product-preserving encryption (ASPE) is distance-recoverable and supports secure and accurate kNN query computation. For details, please refer to [6]. In this paper, we adapt the ASPE to implement the privacy-preserving content-based image retrieval.

3 Our Proposed Scheme

In this section, we present the design of our privacy-preserving content-based image retrieval scheme based on the secure kNN computation. To improve the search efficiency and accuracy, we first describe a method that converts the high-dimensional descriptors to compact binary codes and an asymmetric distance measure. Then we present the details of our privacy-preserving content-based image retrieval scheme.

3.1 BPBC Based Vector Quantization

As introduced in Sect. 2.3, an image can be represented as a $(d * k)$-dimensional vector using VLAD. Then, Euclidean distance between the VLAD descriptors is used for the similarity search. Note that the descriptors should be high-dimensional when the number of clusters k is large and computation over them becomes quite expensive. There are a number of methods to quantize high-dimensional descriptors, including Locality Sensitive Hashing (LSH), Spectral Hashing (SH), and Product Quantization (PQ) [16]. Considering the trade-off between search efficiency and storage requirement [10], in our scheme, we employ the bilinear projection-based binary codes (BPBC) to convert VLAD descriptors into compact binary codes.

For a descriptor $p \in \mathbb{R}^D$, it can be reshaped into a matrix $P \in \mathbb{R}^{d_1 \times d_2}$, $D = d_1 d_2$. Then, two random orthogonal matrices $R_1 \in \mathbb{R}^{d_1 \times d_1}$ and $R_2 \in \mathbb{R}^{d_2 \times d_2}$ are applied to P:

$$H(P) = vec(sgn(R_1^T P R_2))$$

where vec($\cdot$) denotes column-wise concatenation and $H(P)$ is the generated D-dimensional binary string.

Further, reduced-dimension codes can be learned through dimensionality reduction. Random orthogonal projection matrices $R_1 \in \mathbb{R}^{d_1 \times c_1}$ and $R_2 \in \mathbb{R}^{d_2 \times c_2}$ are utilized to produce codes of size $c = c_1 \times c_2$, where $c_1 < d_1$ and $c_2 < d_2$. In this way, we can convert the D-dimensional descriptors of the images to reduced c-dimensional binary codes.

3.2 Asymmetric Distance Based Similarity Measure

At retrieval time, the cloud server determines the similarity between the points in the database and the query point through distance computation. Although both the query image and the images in the cloud can be represented by compact binary codes, to improve the search accuracy, we use asymmetric distance in our scheme. That is, the points in the database are quantized but the query point is not. For the database points $p \in \{-1, +1\}^c$ and a query point $q \in \mathbb{R}^c$, the Euclidean distance between p and q is:

$$d(p, q) = \sqrt{\|p\|_2^2 + \|q\|_2^2 - 2p^T q}$$

Since $p \in \{-1, +1\}^c$, we have $\|p\|_2^2 = c$. And q is on the query side, thus we only need to compute $p^T q$.

In the following, we will adopt the secure inner product scheme to protect the points p in the database and the query point q without revealing the exact distance information between p and q.

3.3 The Privacy-Preserving CBIR

The process of our scheme can be divided into four phases: (1) Key generation phase which is designed to generate the needed secret keys for encryption and decryption of the images; (2) System setup phase which is conducted by the image data owner to generate not only the encrypted image sets that will be outsourced to the cloud server but also the secure indexes of the images; (3) Trapdoor generation phase where the user generates query trapdoors and initiates search sessions with the cloud server; (4) Retrieval phase which is performed by the cloud server to search for the required images and return them to the user in a ranked manner.

Accordingly, our scheme consists of a tuple of four basic polynomial time algorithms as follows:

$\mathrm{K} \leftarrow \mathrm{KeyGen}(1^\lambda)$: given the input security parameter λ, output a set of secret keys K.
$\mathrm{I} \leftarrow \mathrm{BuildIndex}(\Delta, \mathrm{K})$: given the image dataset $\varDelta$ and the secret key K, output the secure index I.
$\mathrm{T} \leftarrow \mathrm{TrapdoorGen}(\mathrm{Q}, \mathrm{K})$: given a query image Q and the secret key K, output the query trapdoor T.
$\mathrm{R} \leftarrow \mathrm{Search}(\mathrm{T}, \mathrm{I})$: given the secure trapdoor T and the index I, output the matched result R.

To enhance the security, we perform a pseudo-random permutation $\pi(\cdot)$ on the binary codes of the data points $p \in \{-1, +1\}^c$. Moreover, we add r dummy dimensions to extend the c-dimensional permuted data points to $(c+r)$-bit. The details are as follows:

Key Generation Phase. The image data owner first runs the KeyGen algorithm to generate the secret keys. The KeyGen algorithm takes as input a security parameter λ and outputs a set of secret keys $K = \{M_1, M_2, S, sk\}$, where M_1 and M_2 are two $(c+r) \times (c+r)$ invertible matrices, S is a $(c+r)$-dimensional bit vector which functions as a splitting indicator, and sk is a symmetric secret key to encrypt the original image data set.

System Setup Phase. Before outsourcing, the owner represents the images as compact binary codes through aforementioned BPBC. Then the owner generates the searchable secure index using the BuildIndex algorithm. We first perform a pseudo-random permutation $\pi(\cdot)$ on p. After that, the permutated vector $\bar{p}$ is extended into $\tilde{p}$ that is $(c+r)$-bit. By choosing t out of r dummy dimensions, the corresponding entries in $\tilde{p}$ are set to 1. And $\tilde{p}$ is further split into two d-dimensional points $\tilde{p_a}$ and $\tilde{p_b}$ in such a way: for $j = 1$ to $c+r$, if the j-th bit of S is 1, $\tilde{p_a}[j]$ and $\tilde{p_b}[j]$ are set to two random numbers so that their sum is equal to $\tilde{p}[j]$, that is $\tilde{p}[j] = \tilde{p_a}[j] + \tilde{p_b}[j]$; if the j-th bit of S is 0, $\tilde{p_a}[j]$ and $\tilde{p_b}[j]$ are set to the same value as $\tilde{p}[j]$, that is $\tilde{p}[j] = \tilde{p_a}[j] = \tilde{p_b}[j]$. Finally, the split data vector pair $\tilde{p}_a$ and $\tilde{p}_b$ is encrypted using M_1^T and M_2^T as $\hat{p}_a = M_1^T \tilde{p}_a$, $\hat{p}_b = M_2^T \tilde{p}_b$. After that, we can outsource the encrypted image data set and the encrypted indexes to the cloud server.

Trapdoor Generation Phase. In order to retrieve the required image data, the user runs the TrapdoorGen algorithm to generate a query trapdoor. Since we use the asymmetric distance, the query image is represented as a reduced c-dimensional descriptor q, which is permuted to be $\bar{q}$ with the same pseudo-random permutation $\pi(\cdot)$. And $\bar{q}$ is extended into $\tilde{q}$ that is $(c+r)$-dimensional. For $j = c+1$ to $c+r$, the entry in $\tilde{q}$ is set to a random number ε_i. Then $\tilde{q}$ is split in the same way as the index except that the split process is the opposite. That is, for $j = 1$ to $c+r$, if the j-th bit of S is 0, $\tilde{q}[j]$ is split into $\tilde{q_a}[j]$ and $\tilde{q_b}[j]$ so that $\tilde{q}[j] = \tilde{q_a}[j] + \tilde{q_b}[j]$; if the j-th bit of S is 1, $\tilde{q_a}[j]$ and $\tilde{q_b}[j]$ are both set to $\tilde{q}[j]$. The split query vectors $\tilde{q}_a$ and $\tilde{q}_b$ are then encrypted using M_1^{-1} and M_2^{-1} as $\hat{q}_a = M_1^{-1}\tilde{q}_a$, $\hat{q}_b = M_2^{-1}\tilde{q}_b$. Finally, the user submits the trapdoor $\{\hat{q}_a, \hat{q}_b\}$ to the cloud server to search for the required image data.

Image Retrieval Phase. After receiving the trapdoor from the user, the cloud server runs the Search algorithm to find out the most similar results against q. Here, we adopt the Euclidean distance to compute similarity between images. Then, for p and q, we have:

$$d(\hat{p}, \hat{q}) = \hat{p}_a^T \hat{q}_a + \hat{p}_b^T \hat{q}_b = \tilde{p}_a^T \tilde{q}_a + \tilde{p}_b^T \tilde{q}_b = \bar{p}^T \bar{q} + \Sigma_{j \in \hat{t}} \varepsilon_j$$

where $\hat{t}$ is the set of the t selected dummy dimensions, and it is different for each index point. In this way, the cloud server can determine the similarity between the query point and the points in the database. And the algorithm selects the top-k secure indexes that have the k biggest matching degree against q and gathers them into a result set R. Finally, the cloud server returns the user with a set of image data that are associated with indexes in R.

4 Security Analysis

In this section, we analyze the security strength of our proposed scheme.

In our scheme, in order to ensure the security of the data stored in the cloud server, the original images are encrypted by standard symmetric encryption algorithm which is semantically secure. As long as the secret key is kept confidential, it can be assumed computationally difficult for the cloud server to infer the original image data. Thus the privacy of the original images are properly protected.

However, during the interaction process between the cloud server and the users, the cloud server can build up access pattern and search pattern based on the recorded trapdoors and search results. Explicitly, access pattern includes the search results corresponding to the query trapdoors, while search pattern includes information describing which queries contain the similar image features.

Therefore, our scheme should guarantee that no additional information beyond the access pattern and the search pattern be leaked to the attackers. We will follow the simulation based adaptive security definition, extended from searchable symmetric encryption(SSE), to investigate the security of our scheme.

Theorem 1. *Our scheme is secure under the known-plaintext model.*

Before proving the Theorem 1, we introduce some notions used in [17].

- *History* is an image data set $\varDelta$, an index set I built from $\varDelta$ and a set of queries $Q = (q_1, \cdots, q_k)$ submitted by the users, denoted as $H = (\varDelta, I, Q)$. H is the plaintext knowledge of the attacker.
- *View* is the encrypted form of H under some secret key sk, denoted as $V(H) = (E_{sk}(\varDelta), E_{sk}(I), E_{sk}(Q))$. Note that the attacker can see $V(H)$ from the cloud server as its ciphertext knowledge.
- *Trace* is the information that the scheme may leak about the history H, denoted as $\tau(H) = \{|E_{sk}(I_i)|, \alpha(H), \sigma(H)\}$, where $|E_{sk}(I_i)|$ is the bit length of the encrypted index; $\alpha(H)$, denoted as the access pattern induced by H, is a tuple $\alpha(H) = (R(q_1), \cdots, R(q_k))$; $\sigma(H)$, denoted as the search pattern induced by H, is a symmetric binary matrix such that $\sigma(H)_{i,j} = 1$, for $1 \leq i, j \leq k$, if q_i and q_j have the similar features, and 0 otherwise.

As defined in [17], given two histories that have the same trace, i.e. $\tau(H') = \tau(H)$, if there is no P.P.T adversary that can distinguish the two views of them, the scheme will be non-adaptively semantically secure.

Proof: Given a history H, there exists a polynomial-time simulator $\mathcal{S}$ that takes $\tau(H)$ as input and outputs a view $V(H') = \{E_{sk'}(\varDelta'), E_{sk'}(I'), E_{sk'}(q')\}$ indistinguishable from an attacker's view. It runs as follows:

- $\mathcal{S}$ initializes an image data set $\varDelta'$ indistinguishable with real $\varDelta$.
- $\mathcal{S}$ generates a visual vocabulary V' indistinguishable with real V.
- $\mathcal{S}$ builds up an index set I' based on V'. I' is indistinguishable with real I.
- $\mathcal{S}$ randomly generates the secret keys: $sk' = \{M_1', M_2', S'\}$ to encrypt I'.
- $\mathcal{S}$ generates a random string to simulate the query trapdoor q' and then operates a random oracle to point at random selected encrypted indexes in I' and reveal the identical identifiers from $\tau(H)$. The identical number of simulated index set randomly assigned from I' are considered to be the simulated results.
- $\mathcal{S}$ sets $V(H') = \{E_{sk'}(\varDelta'), E_{sk'}(I'), E_{sk'}(q)\}$.

The correctness of such construction is easy to demonstrate by querying encrypted q' over encrypted I'. The secure index $E(I')$ and the trapdoor $E(q')$ generate the same trace as the one that the attacker has. Since the image vector is first pseudorandomly permuted with a pseudorandom permutation (PRP) and then encrypted by the secure kNN, due to the properties of the cryptographic primitive (PRP) and the security of kNN, we claim that no P.P.T adversary can distinguish $\{E_{sk}(I), E_{sk}(q)\}$ from $\{E_{sk'}(I'), E_{sk'}(q')\}$. Since Δ is encrypted under standard symmetric encryption such as AES which is considered as pseudorandom and Δ' is randomly generated, the adversary cannot distinguish $E_{sk}(\Delta)$ from $E_{sk'}(\Delta')$. Thus, there is no P.P.T adversary that can distinguish $V(H')$ from $V(H)$ although their underlying histories have the same trace.

Moreover, Yao et al. [18] point out that the secure kNN scheme is susceptible under the chosen plaintext attack. That is, the server can construct d linear equations to derive the coordinates of the data point p through d times observation of the query points and their encryption. However, on the one hand, by using the random dummy dimensions [3], it is not applicable for the cloud server to acquire enough plaintext query information, i.e., the query image vector; on the other hand, the vector is still high-dimensional in our scheme and is randomly permuted before encrypted by the secure kNN method. It is really hard for the attacker to construct such linear equations. Therefore, the attack scenario described in [18] will not make sense.

5 Experimental Evaluation

In this section, we evaluate the performance of our scheme on a real-world image database: Corel Image Set [19], which contains 1000 color images grouped into 10 categories. This database has been used as ground-truth for evaluating color image retrieval and image annotation [5]. We build up our system prototype using the Python language and the computer vision library OpenCV 2.4.11. We deploy the cloud on a server with Intel E5-2620 CPU and 120G RAM, and deploy the user client on a PC with Intel i7-3930 CPU and 16G RAM. The performance of our scheme is evaluated in terms of the search accuracy and efficiency in the process of index construction, trapdoor generation and search operation.

5.1 Search Accuracy

We use the precision and recall rate to evaluate the search accuracy of our proposed scheme. Precision is the percentage of retrieved relevant images over all returned results; recall is the proportion of retrieved relevant images over all the relevant results for the query in the original image data set. Denote TP as true positive, FP as false positive and FN as false negative, then the precision can be calculated as $Precision = \frac{TP}{TP+FP}$, and the recall can be calculated as $Recall = \frac{TP}{TP+FN}$.

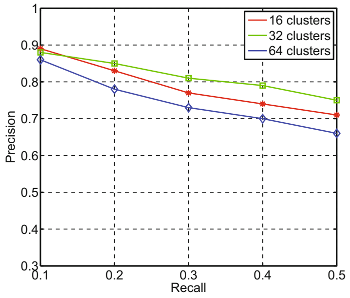

Fig. 2. Search precision and recall with different clusters

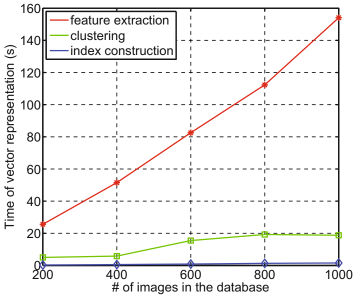

Fig. 3. Time of index construction before encryption

Since VLAD is parametrized by a single parameter k [11], the number of visual words, we first conduct our experiment to verify that excellent results can be obtained even with a relatively small number of k. The existing tool, vlfeat [20], is used to describe the images with SIFT and VLAD descriptors. We choose $k = 16, 32$ and 64 respectively and conduct 50 encrypted queries over the secure indexes. Figure 2 illustrates that comparable search accuracy can be obtained with the three different k values. Since the accuracy depends largely on the techniques such as feature extraction, image representation, and vector quantization, we can utilize more complex image representation techniques combining various features to obtain higher search accuracy.

5.2 Search Efficiency

Index Construction. The index construction process includes feature extraction, clustering, vector representation, vector quantization, and vector encryption. Figure 3 shows that the feature extraction consumes most of the time before encryption and outsourcing compared with clustering and vector representation. Figure 4 shows that the encryption over the descriptors in our scheme is a negligible overhead. Note that the time-consuming parts are the one-off operations and are therefore affordable for the data owner.

Trapdoor Generation. For a query request, the data user needs to instruct the same index as the outsourced images and generates the query trapdoor. The whole process is similar to the index construction, which consists of feature extraction, vector representation and vector encryption. Here, the vector representation is based on the previous clustering step. Through our 50 times of experiment, the average time consumption in the process of trapdoor generation is 1.837 s, which is affordable on the user's side.

Search Operation. After receiving the query trapdoor, the cloud server searches for the top-k images upon the secure index, which includes computing the inner product for all the encrypted indexes in the image data set.

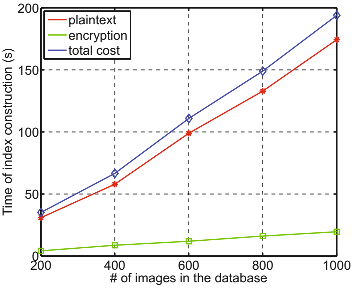

Fig. 4. Time of secure index construction

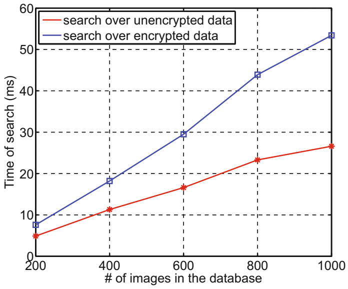

Fig. 5. Time of search over unencrypted and encrypted index

As shown in Fig. 5, image retrieval over the encrypted index consumes comparable time to the unencrypted index and the search time grows linearly with the size of the file set. This is intuitive since we just perform exhaustive computation in our experiment and the search process needs to go over all the files in the dataset before the cloud server can get the final result. Even so, the search can be done within milliseconds, which is comparable to the image retrieval schemes for plaintext.

6 Conclusion

In this paper, we proposed a privacy-preserving content-based image retrieval scheme, which allows the data owner to outsource the images to the untrusted cloud without revealing the privacy of the images. To improve the security and the efficiency, we combined the secure kNN scheme with the BPBC based vector quantization and asymmetric distance based similarity measure. The security analysis and experiment shows the security and efficiency of our scheme. For the future work, we will extend our scheme to support much more complicated application requirements.

Acknowledgment. The work is supported by the National Natural Science Foundation of China under Grant Nos. 61379144 and 61572026.

References

1. Ning, C., Cong, W., Ming, L., Ren, K., Lou, W.: Privacy-preserving multi-keyword ranked search over encrypted cloud data. IEEE Trans. Parallel Distrib. Syst. **25**(1), 222–233 (2014)
2. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: IEEE Symposium on Security & Privacy, pp. 44–55 (2000)
3. Sun, W., Wang, B., Cao, N., Li, M., Lou, W., Hou, Y.T., Li, H.: Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. IEEE Trans. Parallel Distrib. Syst. **25**(11), 71–82 (2013)

4. Wang, B., Yu, S., Lou, W., Hou, Y.T.: Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud. In: 2014 Proceedings IEEE INFOCOM, pp. 2112–2120 (2014)
5. Lu, W., Varna, A.L., Wu, M.: Confidentiality-preserving image search: a comparative study between homomorphic encryption and distance-preserving randomization. Access IEEE **2**, 125–141 (2014)
6. Wong, W.K., Cheung, D.W.l., Kao, B., Mamoulis, N.: Secure knn computation on encrypted databases. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, pp. 139–152 (2009)
7. Chu, W.T., Chang, F.C.: A privacy-preserving bipartite graph matching framework for multimedia analysis and retrieval. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, pp. 243–250 (2015)
8. Zhang, L., Jung, T., Liu, C., Ding, X.: Pop: Privacy-preserving outsourced photo sharing and searching for mobile devices. In: 2015 IEEE 35th International Conference on Distributed Computing Systems (ICDCS), pp. 308–317 (2015)
9. Cui, H., Yuan, X., Wang, C.: Harnessing encrypted data in cloud for secure and efficient image sharing from mobile devices. In: 2015 IEEE Conference on Computer Communications (INFOCOM) (2015)
10. Gong, Y., Kumar, S., Rowley, H., Lazebnik, S.: Learning binary codes for high-dimensional data using bilinear projections. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 484–491, June 2013
11. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3304–3311 (2010)
12. Ferreira, B., Rodrigues, J., Leitão, J., Domingos, H.: Privacy-preserving content-based image retrieval in the cloud. Eprint Arxiv (2014)
13. Xia, Z., Zhu, Y., Sun, X., Qin, Z.: Towards privacy-preserving content-based image retrieval in cloud computing. IEEE Trans. Cloud Comput. 1–11 (2015). doi:10.1109/TCC.2015.2491933
14. Yuan, J., Yu, S., Guo, L.: Seisa: secure and efficient encrypted image search with access control. In: 2015 IEEE Conference on Computer Communications (INFOCOM), pp. 2083–2091 (2015)
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
16. Jégou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Trans. Pattern Anal. Mach. Intell. **33**(1), 117–128 (2010)
17. Curtmola, R., Garay, J., Kamara, S., Ostrovsky, R.: Searchable symmetric encryption: improved definitions and efficient constructions. In: Proceedings of CCS 2006, pp. 79–88 (2006)
18. Yao, B., Li, F., Xiao, X.: Secure nearest neighbor revisited. In: 2013 IEEE 29th International Conference on Data Engineering (ICDE), pp. 733–744 (2013)
19. Wang, J.Z., Li, J., Wiederholdy, G.: Simplicity: semantics-sensitive integrated matching for picture libraries. IEEE Trans. Pattern Anal. Mach. Intell. **23**(9), 947–963 (2001)
20. Vedaldi, A., Fulkerson, B.: VLFeat: an open and portable library of computer vision algorithms (2008). http://www.vlfeat.org/