Received May 25, 2020, accepted June 16, 2020, date of publication June 22, 2020, date of current version July 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003928

Secure Content-Based Image Retrieval in the Cloud With Key Confidentiality

JUNG-SHIAN LI[1], I-HSIEN LIU[1], CHIN-JUI TSAI[1], ZHI-YUAN SU[2], CHU-FEN LI[3], AND CHUAN-GANG LIU[4]

[1]Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, Tainan 701, Taiwan
[2]Department of Information Management, Chia Nan University of Pharmacy and Science, Tainan 717, Taiwan
[3]Department of Finance, National Formosa University, Tainan 632, Taiwan
[4]Department of Applied Informatics and Multimedia, Chia Nan University of Pharmacy and Science, Tainan 717, Taiwan

Corresponding author: Chuan-Gang Liu (chgliu@mail.cnu.edu.tw)

This work was supported in part by the Ministry of Science and Technology under Grant MOST 108-2218-E-006-035- & MOST 109-2218-E-006-014-.

ABSTRACT Owing to the rapid development of cloud services and personal privacy demand, secure cloud storage services and search over encrypted datasets have become an important issue. Recently, the leaking of images such as identification and driver's licenses catches much attention. The trend towards secure computation has been widely discussed, especially asymmetric scalar-product-preserving encryption (ASPE) and homomorphic encryption (HE). Although ASPE have ability to encrypt and determine the similarity between ciphertexts efficiently, it is not a practical methodology due to its assumption that the users are fully trusted in real world and it also may have key leakage problem. Contrary to ASPE, HE can execute addition and multiplication in the encrypted domain and solve key leakage problem. Hence, in this paper, we combine the opinions of HE and ASPE to propose new privacy-preserving content-based image retrieval with key confidentiality scheme against the attacks from data owner, cloud server and users. Our privacy preserving image retrieval scheme is developed under strong threat model that is close to real world. Furthermore, to the best of our knowledge, our work is the first one that developing scheme under the assumption that all the entities involved in privacy-preserving image retrieval system are semi-trusted. Our scheme ensures the confidentiality of key and privacy of query information at the same time. In addition, we provide a lightweight verification to check whether the search query is fake or not. Finally, the experiment results show that the computation overheads and search precision are acceptable at the same time.

INDEX TERMS Secure image retrieval, homomorphic encryption, privacy-preserving computation.

I. INTRODUCTION

With a tremendous growth in digital image applications, we urgently need more storage space than before. Cloud storage is a popular choice to handle a large size data because its cost is much lower than hardware upgrade and infrastructure reorganization. Nowadays, image-based data is getting important in many applications such as face identification, disease detection and object recognition and it usually needs more storage than text-based data. Cloud resource provides not only reduces the storage burden on the local hardware by outsourcing huge image database to the cloud server but also takes advantage of the cloud computation capability for image processing applications. However, the images usually contain personal and confidential information and hence directly outsourcing the image dataset to the cloud will arouse the privacy issue. To protect the sensitive information in images, it is necessary to encrypt images before being uploaded to the cloud. After storing the encrypted image data in cloud, the users will ask the cloud server for the search of the target image data. However, current content-based image retrieval (CBIR) technologies are usually useless for searching encrypted image data, and hence this paper tries to solve the problem of searching encrypted images.

In order to achieve the secure search for encrypted image data, we choose asymmetric scalar-product-preserving encryption scheme [8], which is proposed by Wong. *et al.* in 2009, as main methodology in our scheme. Furthermore, our scheme also applies partially homomorphic encryption to solve the problem of secret key distribution and hence our scheme overcomes the security issue caused by strong threat models. Through combining above schemes, our scheme can

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

address the security problems that usually happen in real world for searching encrypted image data. However, we discover that ASPE can execute kNN query without learning the exact information but it will cause huge computation overheads for comparing all the combinations of two descriptors in large-scale database. Hence, we introduce an index tree, which is based on k-means clustering algorithm, in ASPE, which can improve the search efficiency. In our work, the Partially Homomorphic Encryption library for Python is employed to implement the ASPE without key sharing and it provides secure trapdoor generation at the expense of computation overheads under the strong threat model. Our experiment result shows that the computational error could be within the tolerance 10^{-8}. We will later on describe the detail procedure of privacy-preserving content-based image retrieval.

In our system, the data owner has not to deliver the secret key to the authorized users, because other entities involved in image retrieval are regarded as semi-trusted entity. On the other hand, in our scheme, only data owner is the entity who owns the secret key, the users should ask he/her for permission and trapdoor generation for each search. Usually, the users' search habits are valuable information for the attacks because they may disclose the user's private and personal searching information. In this work, we not only solve the key leakage problem but also forbid the data owner to collect user's search information during the search request phase. The data owner can securely generate the trapdoor without learning anything about the query image. In addition, we proposed a lightweight trapdoor verification to prevent the adversary from faking the search trapdoor and to discard the unauthorized trapdoor when the cloud server receives it from the adversary.

In summary, our system has the following contributions.

- The data owner can securely outsource the images to the cloud server because unauthorized users cannot obtain the information from encrypted images.
- Even though the cloud server tries to get the information of search index by analyzing the data collected during the image retrieval phase, our system ensures that the trapdoor and the secure index are useless for cloud server.
- Besides, we consider the collusion of the malicious users and the cloud server. It is still hard to crack our system, because the data owner keeps the secret key confidential in our scheme.
- Our scheme can provide more secure encryption searching for the image data by combining ASPE and HE schemes.

We also prove that our scheme can resist the strong threat models through the experiments and those results show that our scheme has the acceptable efficacy and high accuracy of secure image search.

In the remainder of our paper, Section 2 reviews several researches related to this paper. In Section 3, we introduce our system architecture and proposed algorithm. Then we give performance evaluation in Section 4 and conclude this paper in Section 5.

II. RELATED WORKS

A. KEYWORD SEARCH

Searchable encryptions have attracted a lot of attention and widely discussed for many years. It changes storage habit nowadays. Most of the existing schemes exploit the secure search index to implement the search function aimed to retrieve the relevant documents from encrypted dataset. Because it is an idea coming from tree structure such as B-tree and *k-d* tree, the search index can retrieve the documents over encrypted dataset efficiently. Furthermore, we protect all non-leaf nodes through encrypting the key value on them and storing the encrypted documents in leaf nodes. Since the all non-leaf nodes are encrypted, only the specific trapdoors generated with the relevant keywords and the corresponding secret key are able to search over the secure index successfully. The adversary cannot easily obtain the index information. Also, the encrypted files stored in leaf nodes would not be accessed unless the user search the secure index with correct trapdoor. Two common algorithms employed to the document retrieval system are shown below.

1) INVERTED INDEX

Inverted index [12], [13] is a famous data structure that maps the documents with the same keywords into the same list. It is commonly used to build searchable encryption system because it provides the ability to handle large-scale datasets.

2) TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY MODEL

Term frequency-inverse document frequency model (TF-IDF) proposed in [15] is a scoring algorithm. It is used to compute the similarity of different documents according to their keywords and rank the retrieval documents with proper weight.

B. IMAGE SEARCH

In this paper, we will introduce the k-NN search, which is the key technology to extend from document search to image search over the encrypted dataset. Before introducing them, we have to elaborate on image representation and find out a way to represent the image in single vector.

Since the computers have no ability to distinguish the contents in images, the image recognition technologies are proposed to solve the problem of computer vision. In recent years, text-based image retrieval (TBIR) and content-based image retrieval (CBIR) [16] are two famous implementations for searching images on databases. Besides, the image retrieval system can obtain the relevant images according to similarity of images.

TBIR is based on keyword search that we mentioned in last section. Instead of using the keywords selected from the documents, it artificially attaches the annotations or keywords

to each image such as landscape, animal and human. Then, we can construct the search index according to the labels like keyword search system does. However, artificially added information results in increase of data size and it is inefficient for human to execute above operation for each image in database. The weakest part of TBIR is the classification results. Even though classifying the same data, the results might be different because the human perception is ambiguous. It is necessary to have an explicit criterion but not intuition to classify the data.

CBIR computes the similarity of images by analyzing the descriptors extracted from images. Local feature and global feature are two types of descriptor and commonly employed in describing the images. Here are the brief introduction of global features and local features as follows:

1) GLOBAL FEATURE

We take global color histogram for example. Global color histogram is a statistics of different channel values in all pixels. RGB and HSV are two common color spaces for representing images. However, it is not accurate in distinguishing those images that have the similarly major color because the color distribution of pixels might be akin, for example, sea and sky. The advantage of using global color histogram is that the complexity of computation is much smaller than local feature. In summary, it is efficient in retrieving relevant images, but the accuracy of retrieval result cannot meet the requirement. Fig. 2 is an example of RGB color histogram. Each pixel in image is 3 bytes of RGB color information and hence one pixel can be represented as (0∼255, 0∼255, 0∼255).

2) LOCAL FEATURE

There are two phases in local feature extraction. First, we have to detect the interest points in image and obtain the positions of them. Second, we should get the descriptors by describing each interest point with its neighborhoods.

Most of the existing applications [32], [33] use local feature as main methodology to represent the image and show great accuracy in object recognition. However, the number of interest points are not fixed in images and the quantity different will influence the search result. In order to reduce the noise, we have to apply weight scale algorithm on the interest points.

In our scheme, we use speeded up robust features (SURF) to describe the images because it improves feature extraction speed which is faster than instance scale-invariant feature transform (SIFT) and provides lower-size descriptor to describe interest point's neighborhood. In addition, it outperforms SIFT for recall in object recognition. Here are the examples of local feature as Fig. 1.

Here, we also discuss some recent researches on secure image retrieval that catches much attention in order to meet the urgent needs of secure image storage and searching in cloud. In 2018, Xia *et al.* [32] developed a secure retrieval framework based on local feature. In their scheme, Earth Mover's Distance is transformed in a manner that CSP can easily evaluate similarity among images. Furthermore, in 2019, Qin *et al.* [34] proposed a secure image retrieval method based on Harris Corner optimization and local sensitive hash (LSH). However, Y. Xu *et al.* [30] discover that the retrieval efficiency is not ideal for large-scale image datasets. Hence, Xu *et al.* proposed secure large-scale image retrieval method in cloud environment. In their work, they use the Hamming embedding algorithm to generate binary signatures of image descriptors. Furthermore, they also combines frequency histogram with binary signatures, which provides a more precise representation of image features in an image and they claimed the retrieval accuracy is improved. In 2019, Ferreira *et al.* [31] also proposed a novel privacy-preserving content-based retrieval scheme. And in their scheme, colour information is encrypted by deterministic encryption techniques, which can enable privacy-preserving image retrieval. In addition, they enable texture information to be encrypted by probabilistic encryption algorithms for better security. Hence, form above scheme, we find several researches try to find more secure privacy-preserving content-based retrieval scheme to improve previous weakness.

FIGURE 1. Local feature: SURF interest points and descriptors.

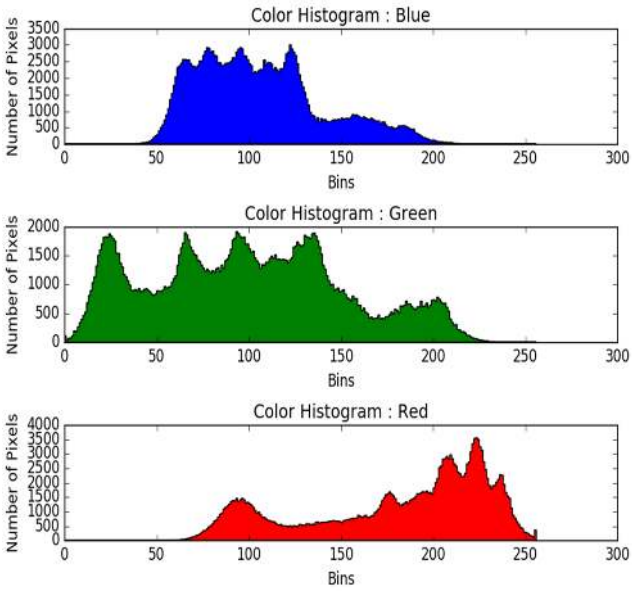

FIGURE 2. Global feature: color histogram of RGB image.

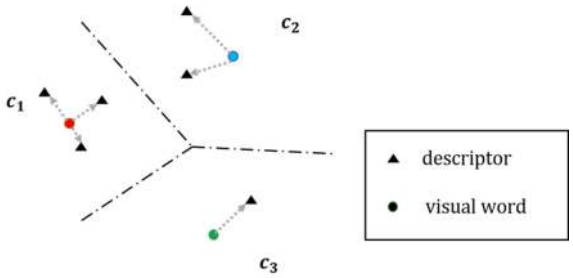

FIGURE 3. Descriptor classification $c_i = NN(x)$.

C. VECTOR OF LOCALLY AGGREGATED DESCRIPTORS

Vector of locally aggregated descriptors [18] (VLAD) is based on local feature and represents the image with single descriptor. In other words, it decreases the searching time significantly because it do not need to search the index repeatedly with massive descriptors. We can regard VLAD as information of feature descriptors that horizontally stacks the difference between visual words and local feature descriptors, so the dimension of VLAD is $D = k \times d$. (Assuming the local feature descriptor is d-dimensional.)

We generate a codebook with local feature descriptors extracted from image dataset at the beginning. There are k centroids $C = \{c_1, c_2 \ldots, c_k\}$ found by k-means algorithm. VLAD is different from BOVW, which only quantizes the descriptors to the closest visual word. Instead of ignoring the quantization error, VLAD records all the difference to reduce the influence of noise. Each local feature descriptor x is associated with its nearest cluster $c_i = NN(x)$ as Fig. 3. We represent the image by VLAD v with indexes $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, d$ standing for visual word and descriptor component, respectively. VLAD accumulates the difference of each descriptor and corresponding visual word. Eq. 1 shows VLAD as follows.

$$v_{i,j} = \sum\nolimits_{x|NN(x)=c_i} x_j - c_{i,j}. \tag{1}$$

Usually, we perform L_2-normalization or power-normalization on VLAD vector v. In this way, the images are distributed in d-dimensional space and the similarity of images can be obtained through different distance metric. The flowchart of computing VLAD descriptor is shown as in Fig. 4.

D. MEAN AVERAGE PRECISION

Most of the existing image search schemes [20]–[22] employ mean average precision (mAP) [19], [28] to analyze the search accuracy. It is an efficient way to measure the practicality of search system. We evaluate the mean of the search precision after a series of searches. Assuming a user searches an image over the datasets and receives 7 relevant images $\{I_1, I_2, I_3, I_4, I_5, I_6, I_7\}$ which are ranked by search system. We get the average precision (AP) with the rank of relevant image in result. r_i is 1 if the I_i is relevant to query image, and 0 otherwise. As we retrieve n images from the database

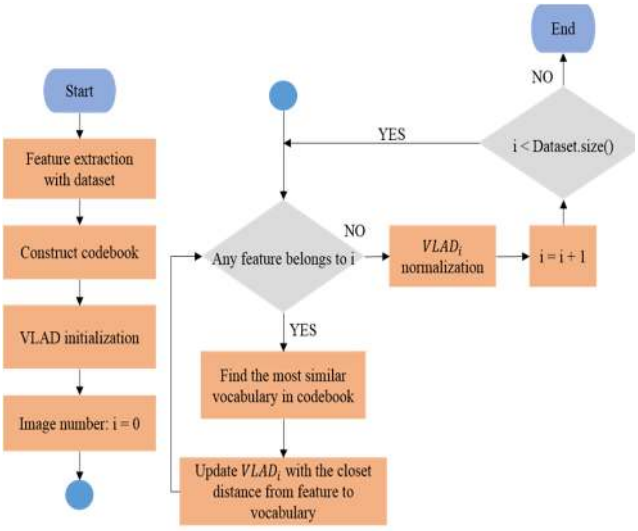

FIGURE 4. Flowchart of representing images with VLAD.

composed of m images, the average precision is obtained just as Eq. 2.

$$v_{i,j} = \frac{1}{\sum_{k=1}^{m} r_k} \sum_{i=1}^{n} r_i \left(\frac{\sum_{j=1}^{i} r_j}{i} \right). \tag{2}$$

The average precision of the search is $AP = (1/1 + 2/3)/(2 + k)$, if the user misses the k relevant images in database and only I_1 and I_3 are correct answer. After searching t times, we can calculated the $mAP = \sum_{i=1}^{t} AP_i/t$.

III. SYSTEM ARCHITECTURE AND PROPOSED ALGORITHM

In this Section, we first introduce the possible adversaries involved in this system and possible security threats for the privacy of the image data, secret key and user query. Then we explain our system architecture and our proposed scheme in details.

A. POSSIBLE ADVERARIES

In this work, the cloud server is regarded as an "honest-but-curious" cloud environment [11], [23], [24]. The cloud server follows the designated protocol, which means he will not do the extra actions for image searching, nevertheless, he tries to obtain the private information by inferring the data that he can collect during the searching service procedure. The assumptions of the data owner and the users in our scheme are quietly different from the most of the existing scheme. The existing schemes [7], [8], [11] assume both of the data owner and the user are trusted except the cloud server. However, it is impractical to implement such system in real world. Hence, we assume both of the data owner and the user are honest-but-curious. Getting over the strong attack becomes the main target since each entity in the privacy-preserving image retrieval system is not fully trusted anymore. Table 1 shows the notation used in this paper.

TABLE 1. Natations used in this paper.

Notations	Definition
p_i	The ith descriptor
q_j	The jth query descriptor
$\hat{p}_i$	The Euclidean norm of p_i
D	The descriptor set representing the corresponding images $D = \{p_1, p_2, p_3 \ldots, p_n\}$ and images is described by each p_i with m-dimensional, where $p_i = \{p_{i1}, p_{i2}, p_{i3} \ldots, p_{im}\}$
$d(p, q)$	Euclidean distance between descriptor p and descriptor q
I	Index tree for image retrieval
I_e	The encrypted index tree for image retrieval
M	secret key
p'_k,	The encrypted descriptor

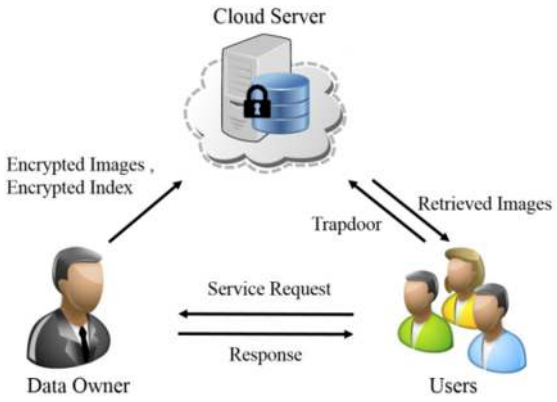

FIGURE 5. System model of our scheme.

B. SECURITY THREATS

Here, we discuss the security threats for the privacy of the image data, secret key and user query in our system environment, respectively, as the following items:

1) PRIVACY OF IMAGE DATA

Due to the encrypted images and secure index are outsourced to the cloud server, they are no longer under data owner's control. We consider two threat models for image privacy in the secure computation [25].

- ***Known Ciphertext Model:*** The cloud server can receives the data from other entities, such as encrypted images, secure indexes and search trapdoor.
- ***Known Background Model:*** The cloud server not only receives the data that we mentioned in Known Ciphertext Model but also knows the detail of index tree construction or encryption methodology.

2) PRIVACY OF SECRET KEY

The secret key is the important part in our scheme and the data owner should keep it confidential. Due to the assumption that the users are not fully trusted in our scheme, the malicious user might collude with the cloud server. However, most of the existing schemes do not consider it. Once the cloud server obtains the secret key form malicious user or careless user, the original index and query detail can be found easily. It will arouse the privacy issue for outsourced data.

3) PRIVACY OF USER QUERY

In our system, the data owner do not distribute the secret key to anyone. In other word, the user has to ask the data owner for generating the search trapdoor in each search. However, sending the image information to the data owner will cause the personal privacy problem, for example, the data owner can guess the user's job or hobby through the statistics of search habits.

In summary, we design our proposed scheme under above possible security risks and ensure our scheme can perform more secure image searching than before. Then, we describe our system architecture in subsection C.

C. SYSTEM ARCHITECTURE

Our search system aims to provide a secure environment that adversary cannot obtain the information without permission for some places requiring more secure environment. For example, medical institution is a place that should have high level security. In this institution, there are millions of medical images need to be stored and those are usually regarded as personal privacy. We do not want the patient's medical records and the system user's search detail to be stolen or disclosed to anyone. Usually, an image retrieval system involves three major entities: the cloud server, the image data owner and the users. The system model is illustrated in Fig. 5 and we will elaborate on each entity.

The system procedure is divided into four phases as follows and we will elaborate on them.

- Setup and key generation
- Secure index building and image data encryption
- Search request
- Image Retrieval

1) SETUP AND KEY GENERATION

In our system, the data owner has to define the secure mechanism to perturb the data in setup phase, for example, adding extra elements into the descriptor of images and permutation function π. The data owner knows the size of perturbed descriptor and ensures the ASPE can execute kNN algorithm correctly during the retrieval phase. Then, the data owner randomly generates an invertible matrix as secret key for ASPE.

2) SECURE INDEX BUILDING AND IMAGE DATA ENCRYPTION

In this phase, we firstly extract the local descriptor from the image database and perform VLAD algorithm to represent each image. Then, we build the tree by recursively conducting k-means algorithm and encrypting all value of non-leaf node with the use of our secure mechanism. In this way, only

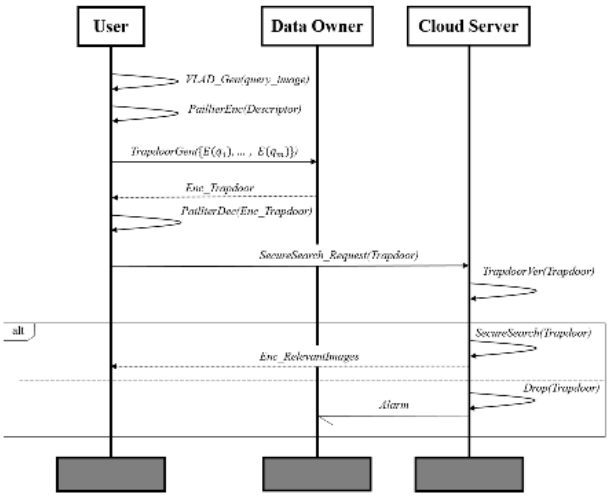

FIGURE 6. Message Sequence chart of searching phase.

the query descriptor which is multiplied by secret key and processed with secure mechanism can access the encrypted images which are stored in leaf nodes through searching the secure index.

3) SEARCH REQUEST

After the data owner transfers the encrypted images and secure index to the cloud server, our system is ready to provide content-based image search service in cloud. Before the user asks for the permission of searching service, he encrypts the query descriptor by Paillier cryptosystem. Although the data owner receives a vector that is composed of ciphertexts, he still can generate the trapdoor securely without knowing anything. When the data owner returns the encrypted trapdoor to the user, the user obtains the trapdoor through decrypting the ciphertext with his own key.

4) RETRIEVAL

Since the trapdoor is generated with the query descriptor and secret key of ASPE, the cloud server can search over the encrypted database with correct trapdoor. In addition, the cloud server is trying to avoid the attacks arising from fake trapdoor. Before retrieving the relevant images, the cloud server will verify whether the trapdoor is generated by the data owner or not. If the cloud server detects fake trapdoor, it stops the service immediately. In last step of image retrieval system, the user has to ask the data owner for the image key because the retrieved data are ciphertexts. Fig. 6 shows the message sequence chart of search request phase and retrieval phase.

Next, we explain our proposed scheme, privacy-preserving image retrieval without key distribution, in our system in details.

D. PRIVACY-PRESERVING IMAGE RETRIVEVAL WITHOUT KEY DISTRIBUTION

Since our scheme represents the images and compute those similarities according to Euclidean distance in d-dimension feature space, the k-nearest neighbor (kNN) query becomes the best algorithm to retrieve top-k relevant data. Previous scheme [8] proposed by Wong. et al also employed kNN query under strong assumptions in their threat models. They solve the privacy issue of cloud computing by asymmetric scalar-product-preserving encryption (ASPE), which can avoid disclosing the distance information and obtain kNN query result by calculating scalar product repeatedly. However, the threat models in their work are still not feasible and impractical. Hence, our scheme tries to utilize homomorphic property for bridging the gap between assumption and reality. Hence, before explaining the procedure of our scheme, we introduce k-NN and the operation of ASPE.

1) k-NEAREST NEIGHBOR ALGORITHM

The k-nearest neighbor algorithm is the simplest machine learning and widely used to analyze data. The object of kNN is classifying the input data according to the labels of top-k nearest neighbors. Since it is a kind of machine learning, we need the training data which is composed of m-dimensional vectors in feature space and label each sample descriptors according to its property before classifying the input data, for example, we use k-means clustering algorithm to label the training data in our work. Notice that k is a user-defined constant representing the number of the nearest neighbors in classifying phase. Based on k-NN algorithm, we can find the the top-k closest descriptors around the query descriptor.

In our scheme, we employ ASPE [18] to implement the secure kNN algorithm and use it to search the relevant images secretly by setting constant k as 1. After training by k-means algorithm, we repeatedly assign the query descriptor to the closest class with the secure index tree during the image retrieval phase and finally find the relevant images in the leaf node. Because the k-NN algorithm only considers the top-k nearest descriptors and assigns the query descriptor a label, computing the accurate distance seems not necessary.

Because of the property of ASPE, we have to compare all combinations of two descriptors in database and figure out the distance relationship among the query descriptors, for example, q_2 is nearer to the query descriptor than q_5. Euclidean distance is a distance metric, which we used to measure the similarity, and the descriptor is similar to the query descriptor if the distance between them are close in feature space. The comparison function of two distance, $d\,(p_1, q)$ and $d\,(p_2, q)$, is shown as Eq. 3.

$$
\begin{aligned}
&d\,(p_1, q) \geq d\,(p_2, q)\,, \\
&\|p_1\|^2 + \|q\|^2 - 2\sum_{j=1}^{m} p_{1j} \cdot q_j \geq \|p_2\|^2 + \|q\|^2 - 2\sum_{j=1}^{m} p_{2j} \cdot q_j \\
&\|p_1\|^2 - \|p_2\|^2 + 2\sum_{j=1}^{m} (p_{2j} - p_{1j})^T \cdot q_j \geq 0 \\
&\|p_1\|^2 - \|p_2\|^2 + 2(p_2 - p_1)^T \cdot q \geq 0. \qquad (3)
\end{aligned}
$$

Now, the comparing function is easy to compute. Instead of comparing exact distances, we can find the descriptor which

is nearer to the query descriptor by calculating Eq. 3. In other words, the asymmetric scalar-preserving encryption can be employed to implement the comparison function without disclosing the sensitive information. In summary, ASPE achieves our aim of taking advantage of cloud computation ability and protecting the outsourced data at the same time.

2) ASYMMETRIC SCALAR-PRODUCT-PRESERVING ENCRYPTION (ASPE)

The procedure of ASPE is divided into three stages as Key Generation, Descriptor Encryption and Search Query. We will elaborate on three stages in this section.

Key Generation: According to the size of descriptor that had been pre-processed, the data owner generates a random invertible matrix as secret key M for the cloud service. Because the secret key should not be disclosed to other entities and the trapdoor is protected by a lightweight verification proposed in this work, our scheme is markedly different from other existing schemes. Furthermore, we don't have to generate two random invertible matrixes to perturb the descriptors with asymmetric random split [8].

Descriptor Encryption: The query descriptor and the descriptors representing the images in database should be encrypted by different encryption functions E_p and E_q, respectively. In our scheme, the encrypted database is searchable only when the cloud server receives the correct trapdoor. The trapdoor is output of encryption function E_p and is useful only when it is constructed with the corresponding secret key, just as Eq. 4 and 5. With this way, the outsourced data in the cloud is secure.

$$E_p\left(\hat{p}, M\right) = p' = M^T\hat{p} \tag{4}$$

$$E_q\left(\hat{q}, M^{-1}\right) = q' = M^{-1}\hat{q}. \tag{5}$$

Search Query: In this stage, ASPE find out the target p which is closer to q. Given two encrypted descriptors p'_1, p'_2 and trapdoor q', the scalar product of $\left(\hat{p}_2 - \hat{p}_1\right) \cdot \hat{q}$ can be obtained through computing $\left(p'_2 - p'_1\right) \cdot q'$ and the encrypted descriptor which is nearer to the query descriptor can be found, just as Eq. 6 and 7. We learn p_2 is closer to the q if the result is positive. Without correct trapdoor and the results of $\left(p'_2 - p'_1\right) \cdot q'$ and $\left(\hat{p}_2 - \hat{p}_1\right)$, the encrypted descriptors are useless for the cloud server.

$$\begin{aligned} p'_2 \cdot p'_1 &= {p'_2}^T p'_1 = \hat{p}_2^T M M^T \hat{p}_1 \neq \hat{p}_2 \cdot \hat{p}_1, \\ \left(p'_2 - p'_1\right) \cdot q' &= \left(p'_2 - p'_1\right)^T q' \\ &= \left(M^T\hat{p}_2 - M^T\hat{p}_1\right)^T M^{-1} q' \\ &= \left(\hat{p}_2 - \hat{p}_1\right)^T M M^{-1} q' \\ &= \left(\hat{p}_2 - \hat{p}_1\right)^T I \hat{q} = \left(\hat{p}_2 - \hat{p}_1\right) \cdot \hat{q} \\ &= 0.5\left(d\left(p_1, q\right)^2 - d\left(p_2, q\right)^2\right). \end{aligned} \tag{6, 7}$$

Although ASPE can execute kNN query without learning the exact information, it will cause huge computation overheads for comparing all the combinations of two descriptors in large-scale database. In this section, we will introduce an index tree which is based on k-means clustering algorithm. It is used to improve the search efficiency. K-means clustering algorithm helps us to classify the data into groups and we could call it vector quantization. The user-defined constant k is the number of cluster centers that are used to represent all the input data. Although k-means clustering algorithm quantizes our data into k clusters and represents the data with cluster centers respectively, the quantization error occurring during the clustering phase might influence the applications result.

Algorithm 1 Building Secure Index

Input: $\{p_i\}$, $1 \leq i \leq n$, invertible matrix M, k-means cluster number K
Output: I_e
Begin
ChildGen $(\{p_i\}_{1\leq i\leq n}, M, K)$
Function ChildGen (vectors $\{p_i\}$, **matrix** M, **int** K) **begin**
  k-means $(\{p_i\}, K) \rightarrow$ K clusters $\{C_k\}_{1\leq k\leq K}$
  for $k = 1; k \leq K; k++$ **do**
   $label = \frac{\sum_{p_i \in C_k} p_i}{N_k}$ // N_k is number of descriptors in cluster kC_k
  $R_k = M^T(label, -0.5\,\|label\|)^T$ // Encrypted Cluster Center
  if $N_k > K$ **then**
    ChildGen $(\{p_i\}_{i\in C_k}, M, K)$;
    end if
  end for
End

Since figuring out the relationships of all the descriptors is time consuming in search phase, we build the hierarchical index tree to reduce the computing burden by recursively applying k-means clustering algorithm (Algorithm. 1). Through clustering the images and corresponding descriptors into different clusters recursively, the images in the same class are likely to be similar and we can approximately filter out the descriptors that is far from query descriptor in search phase.

By combining the k-means clustering algorithm and conducting 1-NN query recursively, only the class which is the nearest to the query descriptor will be consider when the cloud server searches the secure index. This method significantly decreases the computation overhead.

E. ASPE WITHOUT KEY DISTRIBUTION

As the secure index is protected by ASPE, the secret key M should be stored securely. For an attack scenario, the cloud server can decrypt the secure index by collusion with the adversary who has the secret key that $I_{e,i} \in None\ leaf\ node\ in\ I_e$, $E_p\left(I_i, M\right) \cdot M^{-1} = I_i^T M M^{-1} = I_i$. In privacy-preserving computation schemes based on ASPE, the secret key is not for the decryption actually. We regard the secret key as a permission for the cloud service. The users can get the correct result only through correct

trapdoor generated with corresponding secret key. The data owner distributes the secret key to authorized users in existing schemes, however, it might arouse the security issue, for example, the authorized user shares the key to other user who is not allowed to use the cloud service and the cloud server decrypt the encrypted data by collusion with the malicious user.

In this section, we are going to solve the problem of secret key with new service procedure. In our scheme, the user has to ask for permission and trapdoor generation for each search. In addition, the user encrypts the query descriptor to protect the search detail before transferring it to the data owner for trapdoor generation. Although the data is in encrypted domain, the data owner in our scheme still can calculate the matrix manipulation of secret key M and ciphertext without learning anything about the search detail. Only the user who asks for permission can obtain the trapdoor by decrypting the ciphertext which has been securely manipulated by the secret key.

1) PARTIALLY HOMOMORPHIC CRYPTOSYSTEM

Due to the huge computation overheads and intensive communication of the homomorphic encryption [3], it is not practical to implement the privacy-preserving computation nowadays [30]. Instead of using homomorphic encryption as main methodology, we combine the ASPE and homomorphic encryption to develop a feasible scheme and overcome the weak point of APSE.

Paillier encryption [4] is semantic secure and probabilistic encryption. In addition, it is partially homomorphic. It is not strong as fully homomorphic encryption [3] that has both additive and multiplicative homomorphic properties at the same time; nevertheless, it is more sophisticated and more practical. The additive and multiplicative homomorphic property means that we can calculate the encrypted sum or product of two numbers through computing the ciphertexts of corresponding numbers with the public key. The additive homomorphic property of Paillier encryption helps data owner to keep the secret key of ASPE confidential and protect the information of user query via securely multiplication. We briefly discuss the three phases of Paillier encryption and the additive homomorphic property as follows:

Key Generation: Selecting two large and equal length prime numbers p and q. We take $n = p \times q$ and $g = n + 1$ as public key pk. In the other hand, the secret key sk is (λ, μ), where $\lambda = \varphi(n) = (p-1)(q-1)$ and $\mu = \varphi(n)^{-1} mod\ n$.

Encryption: Assuming that the message m is a number where $m \in \mathbb{Z}_n$, we encrypt it with the public key and random $r \in Z_n^*$. The ciphertext of m is $c = g^m \cdot r^n mod\ n^2$ and it is probabilistic cryptography because of the random r. We denote encryption as $E(m, r)$.

Decryption: We can compute the plaintext of c with the secret key sk that $m = L\left(c^{\lambda} mod n^2\right) \cdot \mu mod\ n$. The function $L(x) = (x-1)/n$. We denote decryption as $D(c)$.

Paillier has only one homomorphic property which is additive homomorphic described in [4] as follows:

Given two message m_1 and $m_2 \forall m_1, m_2 \in \mathbb{Z}_n$, we can obtain the sum of two message by decrypting the product of two ciphertext just as Eq. 8.

$$\begin{aligned} & D\left(E(m_1, r_1) \cdot E(m_2, r_2) mod\ n^2\right) \\ &= D\left(g^{m_1} \cdot r_1^n \cdot g^{m_2} \cdot r_2^n mod\ n^2\right) \\ &= D\left(g^{m_1+m_2} \cdot r_3^n mod\ n^2\right) \\ &= (m_1 + m_2) mod n. \end{aligned} \tag{8}$$

With the additive homomorphic property, we can multiply the message m_1 by m_2. (Note that m_2 cannot be ciphertext) we can obtain Eq. 9.

$$\begin{aligned} & D(E(m_1, r_1)^{m_2} mod n^2) \\ &= D(g^{m_1 m_2} \cdot r_1^{nm_2} mod\ n^2) \\ &= D\left(g^{m_1 m_2} \cdot r_3^n mod\ n^2\right) = m_1 m_2 mod\ n. \end{aligned} \tag{9}$$

Since Paillier cryptosystem has the additive homomorphic property as above, we can calculate matrix multiplication with the encrypted messages. It is important to note that only positive integers are allowed to be the elements in matrix, furthermore we must pay attention on the computation result whether it exceeds the public key n or not. It is better to choose large prime numbers in key generation phase or the computation result which is bigger than public key n might be overflow.

2) ASPE WITH KEY CONFIDENTIALITY ALGORITHM

In our image retrieval system, we describe the image with a single vector VLAD that reduces computation overheads in search phase. However, VLAD descriptor, which is composed of decimals, cannot be encrypted directly by Pailler cryptosystem because the message m must be positive integer. In order to achieve the requirement, we transform each component in VLAD descriptor into positive integer by multiplying decimals by the powers of 10 and represent the negative number by shifting method [26]. In our work, the *Partially Homomorphic Encryption library for Python* is employed to implement the ASPE without key sharing. It represents the positive integer as message $m < n/3$ and the negative integer as message $m > 2n/3$. The number located between $n/3 < 2n/3$ is reserved for overflow detection. Then, after computing the result with homomorphic property, the output of matrix multiplication should be scaled back according to the scaling factor 10^r. Our experiment result shows that the computational error could be within the tolerance 10^{-8}.

In the Algorithm. 2, we don't discuss about how it work with scaling and shifting method, however, we show the procedure of computing encrypted trapdoor with a matrix and a column matrix which is composed of ciphertext. In addition, we ensure that the data owner learn nothing about the search query but generate the encrypted trapdoor during the procedure.

Algorithm 2 Matrix Multiplication in Encrypted Domain

Input: encrypted components of query descriptor $\{E(q_1), E(q_2), \ldots, E(q_m)\}^T$, matrix M

Output: encrypted components of trapdoor $E^{(q)} = \{E(M_1 q^T), E(M_2 q^T), \ldots, E(M_m q^T)\}^T$ where M_i is i^{th} row of the matrix M

Begin

for $i = 1; i \leq m; i++$ **do**

initial $E_i^{(q)} = \mathrm{E}^{M_{i1}}(q_1)$

for $j = 2; j \leq m; j++$ **do**

$E_i^{(q)} = E_i^{(q)} \times \mathrm{E}^{M_{ij}}(q_j)$

end for

end for

End

3) TRAPDOOR VERIFICATION

In order to prevent the cloud server from being attacked by fake trapdoor, we take extra steps to verify the source of trapdoor. If the trapdoor fails the verification, the cloud server will drop the search and return nothing to the adversary. Since the data owner generates the trapdoor and secure index, we can secretly hide some information and the cloud server will verify whether the trapdoor meets the format that the data owner designed or it is processed by the adversary.

When the data owner generates the encrypted trapdoor for the user, he selects a non-prime positive integer $r = x \times y$ as an identity for the user and a positive integer ϵ where $\epsilon > 2x$. The data owner extends the encrypted query descriptor $\{E(q_1), E(q_2), \ldots, E(q_m)\}^T$ to $m+2$ dimensional vector $\{E(q_1), \ldots, E(q_m), E(1), E(y)\}^T$ and multiply the matrix M^{-1} by x. Assuming $\dot{q} = xM^{-1}(q, 1, y)^T$, the data owner return $\{E(\dot{q}_1), E(\dot{q}_2), \ldots, E(\dot{q}_{m+2})\}^T$ and $\epsilon \times r$ to the user. Then, the user decrypts the ciphertexts and obtains the trapdoor $\dot{Q} = \{\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_{m+2}, \epsilon \times r\}^T$. On the other hand, we add extra dimension to all non-leaf nodes in index that are as $\dot{P}_i = \{(M^T(p_i, -0.5\|p_i\|^2, \epsilon)^T)^T, -1\}^T$ during the secure index construction phase.

In this way, before searching over the encrypted database through the secure index, the cloud server verify whether the trapdoor had been processed by adversary or not. The details of verifying function are shown as follows.

$$\begin{aligned}\dot{P} \cdot \dot{Q} &= ((M^T(p, -0.5\|p\|^2, \epsilon)^T)^T, -1)^T \\ &\quad \cdot ((xM^{-1}(q, 1, y)^T)^T, \epsilon \times r)^T \\ &= x\left(p^T q - 0.5\|p\|^2 + \epsilon y\right) - \epsilon \times r \\ &= -0.5x\left(d^2(p, q) - \|q\|^2\right) + \epsilon(xy - r). \quad (10)\end{aligned}$$

Since $-1.5x \leq -0.5x\left(d^2(p, q) - \|q\|^2\right) \leq 0.5x$, the cloud server finds out that the trapdoor is not constructed by the data owner and stops the service if the result of $\dot{P} \cdot \dot{Q}$ is not located between $-1.5x$ and $0.5x$. Even though the fake trapdoor passes the verification by colluding with the cloud server, the adversary still cannot know the detail of original query descriptor because the trapdoor is probabilistic encryption.

TABLE 2. Hardware and software specifications.

CPU	Intel Core i5-3210M CPU @ 2.50GHz, 4 cores
RAM	8 GB DDR3
Operating System	Linux Ubuntu 16.04 (64 bits)
Language	Python2, Python3
Libraries Used	GMPY, Numpy, Sklearn & PHE

By perturbing the trapdoor with matrix and identity r, the trapdoor corresponds to a specific identity, which increases the difficulty of analyzing trapdoors for attacking the system.

IV. PERFORMANCE EVALUATION

In this section, we implement the privacy-preserving content-based image retrieval with key confidentiality and design a series of experiments to evaluate the practicality of our scheme. In addition, we discuss the computation overheads for overcoming the strong threat models and prove our scheme ha an acceptable cost to achieve our goal.

A. SIMULATION SETUP

We build the experiment on Linux Ubuntu system with 2.5GHz Intel Core i5 CPU and 8GB of RAM. All the implementations and experiments are using Python with INRIA Holidays dataset [29] and 10,000 images crawled from Internet. General Multi Precision Python (GMPY), which is C-coded Python modules, is employed to implement the homomorphic encryption in our scheme. The specifications of our hardware and software are shown as follows.

B. PERFORMANCE

1) ACCURACY COMPARISON

In this section, we evaluate the practicality of search systems by mean average precision (mAP), which is introduced in Section II and the experiments will be emphasizing on the influence of different descriptor, principal components analysis (PCA) and codebook size.

Since we use SURF to describe the interest points, Table 3 shows that, in representing the image with VLAD, our scheme is more accurate than ALIDC [18] which describes the interest points by PCA-SIFT. In addition, our scheme's accuracy is close to fisher descriptor. We could regard VLAD as simple version of fisher vector and it is more efficient during the training phase.

Fig. 7 shows the mAP with varying codebook size and dimension reduction. It is instinctive that the larger size of codebook will provide higher accuracy, because it can represent the image more specifically. In our experiment result, applying the PCA on the VLAD descriptors might increase the search accuracy while the codebook size is getting larger. The reason is that the higher dimension of descriptor generally might have more redundant components, which are

TABLE 3. Comparison of accurancy in different schemes.

Schemes	*Descriptors*	*K*	*D*	$D' = D$	$D' = 2048$	$D' = 512$	$D' = 128$
SEISA	Fisher	32	4096	61.7	57.4	55.1	42.8
ALIDC	Fisher	64	4096	59.5	60.7	61.0	56.5
ALIDC	VLAD	64	4096	55.6	57.6	59.8	55.7
Our scheme	Fisher	64	4096	58.4	59.5	62.1	60.6
Our scheme	VLAD	64	4096	60.2	60	61.5	60.9

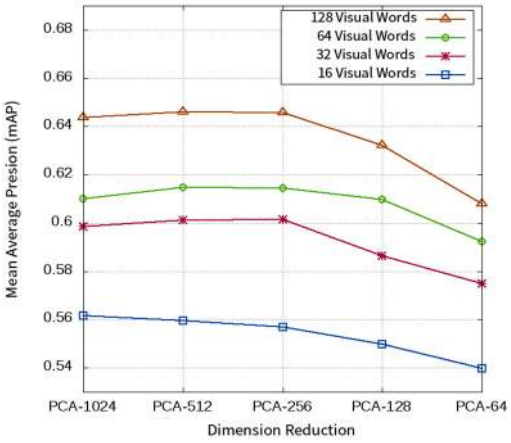

FIGURE 7. Search on Holidays dataset + 10k images crawled from Internet with varying size of codebook and dimension reduction.

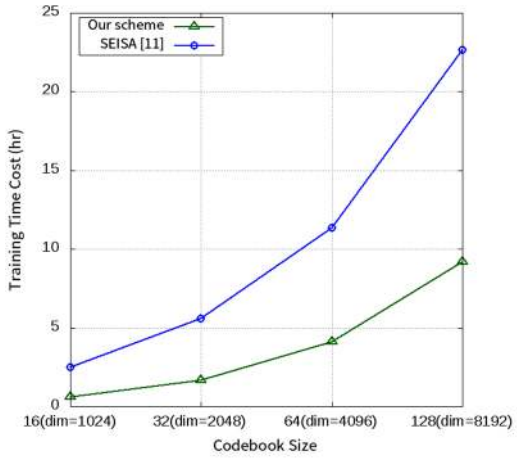

FIGURE 8. Time cost of training.

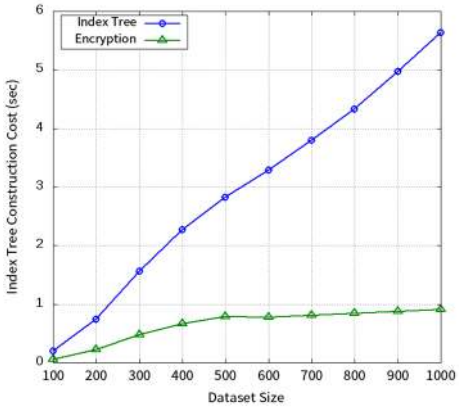

FIGURE 9. Time cost of secure index construction.

the noise during the search phase. Our scheme employ PCA to remove the redundant components, so the accuracy is improved slightly.

2) PERFORMANCE OF SECURE SEARCHABLE INDEX

To support privacy preserving image retrieval, secure searchable index is the important part of our scheme. In this section, we do not consider the local feature, because the comparison of SURF, SIFT and PCA-SIFT is not our focus point and it has been widely discussed.

As we mentioned before, VLAD is good at training phase, because our scheme use k-means algorithm, which is faster than fisher vector constructed by the Gaussian mixture model (GMM), to find the most representative bag of visual words. According to the experiments of Fig. 8 and Table 2, our scheme can build the secure search index faster with only sacrificing a little accuracy. After we represented the images by VLAD descriptor, it is ready to build the secure search index. In our scheme, the index tree is built through recursively executing the k-means algorithm, which the overhead is $O(kNT)$. Fig. 9 shows that the time cost of index building is linear growth with the size of dataset that we set $k = 16$ and number of iteration $T = 700$. Note that the experiment of Fig. 9 do not contain the 10k extra images. The datasets in Fig. 9 are randomly selected from holidays dataset. Fig. 10 show that the time cost of different jobs during index construction.

Since our system perturbs the VLAD descriptor through multiplying a vector by a random invertible matrix, the overhead of encryption is $O(N^2)$ and the size of codebook is the decisive factor to the time cost of encryption. Fig. 11 shows the improvements of different dimension reduction and we notice that conducting PCA-512 on our VLAD descriptor can achieve the better accuracy and reduce the time cost significantly.

3) PERFORMANCE OF SECURE IMAGE RETRIEVAL

Different from the existing schemes, our scheme is developed under the strong threat models. Table 4 is the comparison of various schemes. SEISA and EPCBIR are proposed in the past three year and they still assume all the entity in system are trusted except the cloud server.

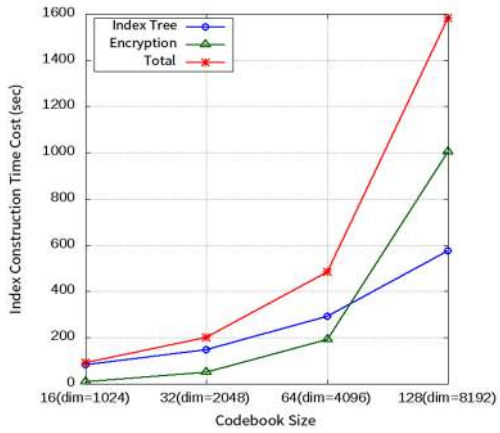

FIGURE 10. Time cost of secure index construction with varying codebook size.

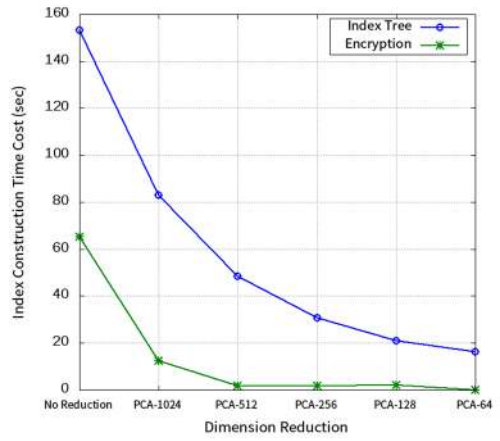

FIGURE 11. Time cost of secure index construction with varying dimension reduction (codebook size : 32).

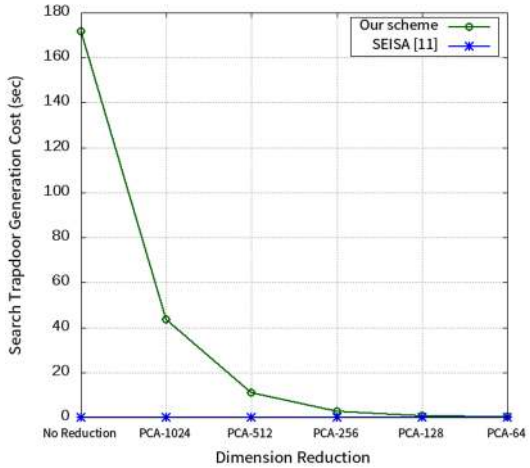

FIGURE 12. Time cost of trapdoor generation(codebook size : 32).

In order to generate secure trapdoor, we utilize the homomorphic property to compute the matrix multiplication, which usually leads to some additional computation overheads. Hence, Fig. 12 shows this circumstances. However, if we employ dimension reduction – PCA-512, we find that our scheme has similar computation cost as SEISA.

TABLE 4. Computation of assumption.

Scheme	Cloud Server	Data Owner	User
SEISA	**Semi-trust**	Trust	Trust
EPCBIR	**Semi-trust**	Trust	Trust
Our scheme	**Semi-trust**	**Semi-trust**	**Semi-trust**

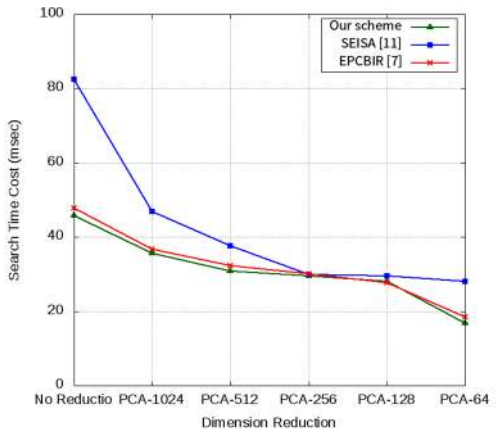

FIGURE 13. Time cost of searching the relevant images over encrypted dataset containing 10k images (codebook size : 32).

Fig. 13 shows the search efficiency of different schemes. Since SEISA extend the fisher vector with extra components to support the access control, it spends much time to retrieve images. In the other hand, our scheme is faster than EPCBIR because the cloud server does not need to recover the split trapdoor which is a method to perturb the trapdoor in [7]. Hence, we can claim that our scheme actually achieve better searching time cost.

V. CONCLUSION

In this work, in order to achieve secure search for the encrypted image retrieval system, we develop a new privacy preserving image retrieval system in which we combine ASPE and HE schemes. Furthermore, to the best of our knowledge, our proposed scheme is the first work that assuming that all the entities are semi-trusted in this system. However, ASPE implements kNN for searching dataset, which also cause serious computation overhead. In order to improve the performance of the search time, k-means algorithm is applying in ASPE to simplify the descriptors of large-scale database containing over 10k images. Furthermore, our proposed scheme also utilize HE scheme to keep the secret key of ASPE confidential. In our scheme we also apply trapdoor verification in searching phase to confirm the validation of trapdoor. Hence, through combination of ASPE and HE, our scheme provide a more secure image retrieval in cloud.

In our scheme, each image is represented by the single vector. However, for the high dimensional descriptor, it leads

to huge computation overheads, especially in executing the encrypted function. Hence, in the future, we continue this study to propose a privacy-preserving image retrieval system based on both local feature and global feature. We try to develop a decision scheme to optimize lower codebook size to improve the system efficiency and ensure the high accuracy meanwhile.

REFERENCES

[1] W. Lu, A. Swaminathan, A. Varna, M. Wu, and D. Electrical, "Enabling search over encrypted multimedia databases," *Proc. SPIE*, vol. 7254, Feb. 2009, Art. no. 725418.

[2] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 17–22.

[3] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. 41st Annu. ACM Symp. Symp. Theory Comput. STOC*, 2009, pp. 169–178.

[4] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 1999, pp. 223–238.

[5] W.-T. Chu and F.-C. Chang, "A privacy-preserving bipartite graph matching framework for multimedia analysis and retrieval," in *Proc. 5th ACM Int. Conf. Multimedia Retr. ICMR*, 2015, pp. 243–250.

[6] H. Cui, X. Yuan, and C. Wang, "Harnessing encrypted data in cloud for secure and efficient image sharing from mobile devices," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2015, pp. 2659–2667.

[7] K. Huang, M. Xu, S. Fu, and D. Wang, "Efficient privacy-preserving content-based image retrieval in the cloud," in *Proc. Int. Conf. Web-Age Inf. Manage.*, Nanchang, China, 2015, pp. 28–39.

[8] W. K. Wong, D. W.-L. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases," in *Proc. 35th SIGMOD Int. Conf. Manage. Data SIGMOD*, 2009, pp. 139–152.

[9] K. Chen, G. Sun, and L. Liu, "Towards attack-resilient geometric data perturbation," in *Proc. SIAM*, Apr. 2007, pp. 78–89.

[10] B. Yao, F. Li, and X. Xiao, "Secure nearest neighbor revisited," in *Proc. IEEE 29th Int. Conf. Data Eng. (ICDE)*, Apr. 2013, pp. 733–744.

[11] J. Yuan, S. Yu, and L. Guo, "SEISA: Secure and efficient encrypted image search with access control," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 2083–2091.

[12] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," in *Proc. 13th ACM Conf. Comput. Commun. Secur. CCS*, 2006, pp. 79–88.

[13] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in *Proc. ACM Conf. Comput. Commun. Secur. CCS*, 2012, pp. 965–976.

[14] Y. Suzuki, M. Mitsukawa, and K. Kawagoe, "A image retrieval method using TFIDF based weighting scheme," in *Proc. 19th Int. Conf. Database Expert Syst. Appl.*, Sep. 2008, pp. 112–116.

[15] C. Sasarak, K. Hart, and R. Pospesel, "A multimodal Web interface for math search," in *Proc. HCIR*, Oct. 2012, pp. 1–4.

[16] C. D. Manning, P. Raghavan, and H. Schutze, *An Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[17] Sivic and Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1470–1477.

[18] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3304–3311.

[19] Y. Yue, T. Finley, F. Radlinski, and T. Joachims, "A support vector method for optimizing average precision," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. SIGIR*, 2007, pp. 271–278.

[20] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3384–3391.

[21] M. Douze, A. Ramisa, and C. Schmid, "Combining attributes and Fisher vectors for efficient image retrieval," in *Proc. CVPR*, Jun. 2011, pp. 745–752.

[22] Perronin, J. Sanchez, and T. Mensink, "Improving the Fisher kernel for large-scale classification," in *Proc. ECCV*, Berlin, Germany, Sep. 2010, pp. 143–156.

[23] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 525–533.

[24] B. Ferreira, J. Rodrigues, J. Leitao, and H. Domingos, "Privacy-preserving content-based image retrieval in the cloud," in *Proc. IEEE 34th Symp. Reliable Distrib. Syst. (SRDS)*, Sep. 2015, pp. 11–20.

[25] K. Liu, C. Giannella, and H. Kargupta, "An attacker's view of distance preserving maps for privacy preserving data mining," in *Proc. PKDD*, Berlin, Germany, 2006, pp. 297–308.

[26] Y. Zhu, Z. Huang, and T. Takagi, "Secure and controllable k-NN query over encrypted cloud data with key confidentiality," *J. Parallel Distrib. Comput.*, vol. 89, pp. 1–12, Mar. 2016.

[27] A. Turpin and F. Scholer, "User performance versus precision measures for simple search tasks," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. SIGIR*, 2006, pp. 11–18.

[28] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometry consistency for large scale image search," in *Proc. ECCV*, Marseille, France, 2008, pp. 12–18.

[29] Z. A. Abduljabbar, H. Jin, A. Ibrahim, Z. A. Hussien, M. A. Hussain, S. H. Abbdal, and D. Zou, "SEPIM: Secure and efficient Private Image Matching," *Appl. Sci.*, vol. 6, no. 8, pp. 1–21, Jul. 2016.

[30] Y. Xu, X. Zhao, and J. Gong, "A large-scale secure image retrieval method in cloud environment," *IEEE Access*, vol. 7, pp. 160082–160090, 2019.

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

[32] Z. Xia, Y. Zhu, X. Sun, Z. Qin, and K. Ren, "Towards privacy-preserving content-based image retrieval in cloud computing," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 276–286, Jan. 2018.

[33] 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.

[34] J. Qin, H. Li, X. Xiang, Y. Tan, W. Pan, W. Ma, and N. N. Xiong, "An encrypted image retrieval method based on harris corner optimization and LSH in cloud computing," *IEEE Access*, vol. 7, pp. 24626–24633, 2019.

JUNG-SHIAN LI received the Ph.D. degree in computer science from the Technical University of Berlin, Germany, in 1999. He is a Full Professor with the Department of Electrical Engineering, National Cheng Kung University (NCKU), Taiwan. He is the Director of the TWISC, NCKU. His research interests include network protocol design, security, and network management.

I-HSIEN LIU received the Ph.D. degree in computer and communication engineering from National Cheng Kung University (NCKU), Taiwan, in 2015. He is a Researcher Fellow of the TWISC, NCKU, where he is also a Researcher Fellow of the Department of Electrical Engineering. His research interests are cloud security, wireless networks, and reliable transmission in mobile networks.

CHIN-JUI TSAI was born in Taipei, Taiwan, in 1992. He received the B.S. degree in communications engineering from Yuan Ze University, Taoyuan, Taiwan, in 2015, and the M.S. degree in computer and communication engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, in 2017.

ZHI-YUAN SU is a Professor with the Department of Information Management, Chia Nan University of Pharmacy and Science. His researches focus on medical information and decision support systems.

CHU-FEN LI received the Ph.D. degree in information management, finance, and banking from Europa-Universität Viadrina Frankfurt, Germany. She is an Associate Professor with the Department of Finance, National Formosa University, Taiwan. Her current research interests include intelligence finance, e-commerce security, financial technology, and the IoT security management.

CHUAN-GANG LIU received the M.S. and Ph.D. degrees in electrical engineering from National Cheng Kung University. He is an Associate Professor with the Department of Applied Informatics and Multimedia, Chia Nan University of Pharmacy and Science. His research interests include wireless networks, network security, and performance analysis.