Privacy-Enhanced Similarity Search Scheme for Cloud Image Databases

Hao LIU††, Nonmember and Hideaki GOTO††, Member

SUMMARY The privacy of users’ data has become a big issue for cloud service. This research focuses on image cloud database and the function of similarity search. To enhance security for such database, we propose a framework of privacy-enhanced search scheme, while all the images in the database are encrypted, and similarity image search is still supported.

key words: privacy-enhanced search, cloud service, content-based image retrieval (CBIR), homomorphic encryption, searchable encryption

1. Introduction

With higher access speed and more various functions, cloud services become common tools for both commercial customers and other general users. When considering the data outsourced to the cloud, multimedia data, especially images, constitute a high proportion. Compared with traditional text data, images often contain more privacy-sensitive information such as facial images in a facial recognition system. Thus, it is quite necessary to reassure the users about the privacy of outsourced images as well as maintaining the functionality of the system.

One trivial way to solve the privacy problem can be generally summarized as searchable encryption which allows one to do query over encrypted data [1]. So far, keyword based searchable encryption works well for text files, for example emails. However, for image retrieval, the scheme has not been practical yet, because it is very costly to assign a precise keyword to every image, without leaking any information of the raw image to the cloud server providers.

In this paper, we study the data privacy issue for image similarity search under the cloud environment. The techniques of image feature extraction, homomorphic encryption and secure multi-party computation are used to achieve secure and efficient data retrieval over encrypted images. The proposed scheme allows the cloud servers to calculate the similarity between images while all the raw information of the images is kept secret to the cloud service providers.

There is actually not very much research focusing on this particular area. A similar research to this work is Hsu’s [2], which proposed a secure SIFT image matching scheme over encrypted data. However, the data owner needs to encrypt and decrypt every pixel in the image using the Paillier cryptosystem. In Barni’s work [3], they use a similar architecture for fingerprint recognition. In their scheme, the Paillier cryptosystem is also used to calculate the distance between two fingerprint samples. In order to further judge whether samples are similar, it requires data exchange with the user, which will increase the user’s computational burden. In Yasuda’s work [4], they developed a homomorphic encryption scheme for calculating Euclidean distance. Although the value of distance can be calculated, the comparison between two values, which is required in the process of image searching, is not implemented by their algorithm.

The rest of this paper is organized as follows: Section 2 describes some techniques needed for image feature extraction and data encryption. Section 3 presents a new privacy-enhanced similarity image search scheme. In Sect. 4, we introduce a scheme to increase system performance. Finally, a conclusion is made and our future work is introduced.

2. Image Retrieval in the Encrypted Domain

2.1 Image Feature Extraction

For image similarity search and Content-based Image Retrieval (CBIR) applications [5], one of the common techniques is based on feature extraction. Feature extraction algorithms can be used to detect some characteristics that do not change in viewing conditions (scale, orientation, contrast, etc.). Thus, these image features, represented by some kinds of feature descriptors, contain the feature of an image and can be used for matching with other similar images. A feature descriptor is usually represented by a set of vectors. In this way, an image can be mathematically described and matching with similar images can be done by calculating the distance between feature descriptors. There exist many feature extraction algorithms. In this research, we choose to use the Speeded-Up Robust Features (SURF) approach which is efficient and can provide acceptable accuracy as well, especially when combined with other techniques [6].

Under SURF, a feature descriptor is represented by a 64-dimensional vector. Let $A$ denote a query image and $B$ denote an image stored in the database. They both can be represented by a set of feature descriptors as $A = (a_1, a_2, ..., a_m)$ and $B = (b_1, b_2, ..., b_n)$, respectively, where $m$ and $n$ are the number of descriptors of $A$ and $B$, respectively. Let $a_i$ be the $i$-th descriptors in $A$ and $b_j$ be the $j$-th descriptors in $B$. The key idea is to...
\( b_j \) be the \( j \)-th descriptors in \( B \). The Euclidean distance between two feature descriptors \( a_i = (d_{i1}, d_{i2}, ... d_{i64}) \) and \( b_j = (b_{j1}, b_{j2}, ... b_{j64}) \) is defined as follows:

\[
distance_{a_i, b_j} = \sqrt{\sum_{k=1}^{64} (d_{ik} - b_{jk})^2}
\]

(1)

To simplify the calculation, in our schemes we use the square of the Euclidean distance because it provides the same results for matching similar images.

Then the distance between every combinations of two feature descriptors in \( A \) and \( B \) can be calculated. The distance of \( a_i \) to the image \( B \) is defined as the minimum distance between and all the feature descriptors of \( B \):

\[
distance_{a_i, B} = \min(\text{distance}_{a_i, b_k}) \quad (1 \leq k \leq n)
\]

(2)

With all the \( \text{distance}_{a_i, B} \) \((1 \leq i \leq m)\), it is possible to find out the most \( K \) similar images in the database. For example, we can calculate the variance of \( \text{distance}_{a_i, B} \) \((1 \leq i \leq m)\). After calculating the variance between image \( A \) and all the images in the database, we can then sort all the variance into ascending order and pick up the first \( K \) results.

2.2 The Paillier Cryptosystem

To solve the privacy problems for cloud computing, there is a technique called homomorphic encryption. It allows computations to be carried out on ciphertext, which means the cloud server can process user’s encrypted data without decrypting them. Based on the calculation required in the proposed system, the Paillier cryptosystem [7] is chosen.

The Paillier cryptosystem is a probabilistic additive homomorphic cryptosystem, which means that one can compute the encrypted sum of two numbers by only knowing the ciphertext of these two numbers. For two numbers \( m_1 \) and \( m_2 \), this can be described as follow:

\[
D(E(m_1, r_1) \times E(m_2, r_2) \mod n^2) = (m_1 + m_2) \mod n
\]

(3)

Here \( E(m) \) and \( D(m) \) denote the encryption and decryption functions, respectively, and \( n \) is the product of two large prime numbers. The two numbers \( r_1 \) and \( r_2 \), which are both randomly generated for every encryption, ensure that the Paillier cryptosystem is probabilistic.

3. Privacy-Enhanced Similarity Image Search

3.1 System Architecture

The main goal for the proposed system is to perform similarity image search while the raw feature descriptors of the images are encrypted. The reason to encrypt all the feature descriptors is that they can be used to identify a particular image. If they were not encrypted, the cloud operator could use them to easily guess the content of the user’s images.

The proposed system is mainly designed for user devices with limited computational power. As the approach mentioned in [3], it requires heavy data transmission between the user terminal and the cloud server, and complex decryption process by the user. To avoid such problems, in the proposed system, there is an extra server to take the part of the user’s task. Additionally, in order to prevent collusion between the two servers, some other security control techniques should be applied to a practical system besides encryption, for example user authentication and authorization.

As shown in Fig. 1, in this system, there are three types of entities: the data owner, the data user and the cloud servers. The data owner holds a large set of images to be outsourced to the cloud. Similar to other searchable encryption systems, the data owner encrypts the images and generates searchable index for each image before outsourcing the images to the cloud. In the query phase, the data user submits an encrypted query to the cloud servers. Then for privacy-enhancing purpose, the cloud servers should be able to run a special searchable encryption scheme to find the images in the cloud that are similar to what the data user has provided. Based on the similarity between the query image and the result images, the cloud servers return the encrypted most \( K \) similar images to the data user. The authorized data users can decrypt the requested images, and they can further process those images to pick out the most wanted results.

3.2 System Design Details

The whole process can be divided into two phases: the setup phase and the query phase.

\textit{1) Setup Phase}

In this phase, the data owner needs to encrypt the image, and also generates the corresponding index. Then the index and the image are uploaded to the cloud server together.

\textit{Step 1: Feature Extraction.} For each image, the data owner extracts all its feature descriptors and form a feature vector \( F = (f_1, f_2, ... f_m) \), where \( f_i \) is the \( i \)-th descriptor of the image. Currently, the features are extracted with SURF. As in SURF every descriptor has 64 dimensions, it can be denoted as \( f_i = (f_{i1}, f_{i2}, ..., f_{i64}) \).

\textit{Step 2: Key Generation.} First, \( key_{img} \) is generated for
encrypting the images. There exist many techniques for image encryption. However, in this paper, it is not our main focus, so here we assume the image encryption is strong enough to keep the outsourced images confidential against the cloud server provider. Then, key\textsubscript{index}, which is the key pair for the Paillier encryption, is generated for encrypting the feature vector. It contains a public key key\textsubscript{public} and a private key key\textsubscript{private}.

**Step 3: Data Encryption.** The image is first encrypted with key\textsubscript{img}. Then every element of the feature descriptor is encrypted with key\textsubscript{private}:

\[
 f_i = (f_{i1}^r, f_{i2}^r, \ldots, f_{ir}^r)
\]

Here, every element \( f_{ik}^r = E(f_{ik}, \text{key}_{\text{private}}) \), which is the Paillier encryption of \( f_{ik} \) with key\textsubscript{private}. All the encrypted descriptors of an image are combined together to form an index for that image.

**Step 4: Data Upload.** The encrypted image and the corresponding encrypted index are uploaded to the cloud server. Necessary keys also need to be shared, which will be further introduced in the next subsection.

2) **Query Phase**

In this phase, only the authorized data users are able to get the key\textsubscript{private} and key\textsubscript{img} from the data owner, in order to generate query and decrypt the returned image.

**Step 1: Secure Query Generation.** Similar to the index generation in the setup phase, the data user first extracts the feature vector of the query image, then encrypt it with the private key key\textsubscript{private}.

**Step 2: Image Matching.** The secure query request is sent to the cloud server for matching. After receiving the query request, the cloud servers calculate the Euclidean distance between feature descriptors of the query image and all the images stored in the database. All the values of distance need to be compared in order to judge whether two images are similar.

However, under the Paillier cryptosystem, only \( a_{ik} - b_{jk} \) can be calculated. It is impossible to further calculate the Euclidean distance without decrypting \( a_{ik} \) and \( b_{jk} \). To solve this problem, we have introduced an extra cloud server into the system. This scheme will be explained in the next subsection.

**Step 3: Result Retrieval.** The \( K \) encrypted images in the database with the \( K \) smallest distance are selected as the query result. These encrypted files are sent to the data user and then by using the key\textsubscript{img}, the images can be revealed.

### 3.3 Image Match with Secure Multi-Party Computation

With an additional cloud server, the whole query scheme now is processed as shown in the Fig. 2.

For every query, Server 1 now only calculates intermediate results:

\[
 M_{i,j} = ((a_1^r - b_1^r), (a_2^r - b_2^r), \ldots, (a_{w}^r - b_{w}^r))
\]

\[
 M_i = (M_{i,1}, M_{i,2}, \ldots, M_{i,w})
\]

where \( w \) is the number of feature descriptors of an image stored in the database. The intermediate results and the corresponding image ID of all images are then sent to Server 2. The calculation and transmission of intermediate result (step 2 and 3 in Fig. 2), and the calculation of distance (step 4 in Fig. 2) can be concurrently processed.

In order to decrypt the intermediate results, Server 2 needs to obtain key\textsubscript{private}. Thus, all the keys that should be shared during the setup phase are shown in Fig. 3. Since the private key is considered to be more classified, Server 2 should be more trustworthy.

After decryption, the distance can be calculated and then Server 2 can judge whether the two images are similar. Then it will return the image ID of the most similar \( K \) images to Server 1.

In our system model, the network between the two cloud servers is assumed to be much faster and more reliable than the network between the data users and the cloud servers. Additionally, the cloud servers are assumed to be honest-but-curious, which here particularly means that Server 2 will not exchange the key\textsubscript{private} with Server 1. Considering the security of such multi-party computation scheme, besides functionality, the confidentiality of data is very important. For the proposed scheme, the value of the encrypted searchable index and the query is kept secret from Server 1. This is ensured by the strength of the Paillier crypt-
They yield values between 0 and 2048 bits. After adding a common offset by using a 2048-bit key in one Paillier ciphertext. Figure 4 shows an example for group encryption.

In the basic system architecture, data transmission between two servers becomes a bottleneck to the system’s time performance. It is mainly because that for every image, there will be hundreds of feature descriptors and every feature descriptor has 64 vectors needed to be encrypted separately. We will propose a scheme to reduce this data transmission.

In order to retain a certain security level, the length of the key used for encryption is recommended to be at least 2048 bits [9]. Here we try to pack all 64 vector elements into one encryption. The elements of the feature vectors obtained by SURF in OpenCV are stored as 7-digit signed integer values. After adding a common offset to these values, they yield values between 0 and 2 × 107, which take 25 bits at maximum. Thus, it is possible to encrypt all the 64 vector elements, together with one extra bit for each element, by using a 2048-bit key in one Paillier ciphertext. Figure 4 shows how the scheme works:

1. During data outsourcing, for a feature descriptor, all the vector elements are simply joined together in their original order, while adding an extra bit “0” between adjacent elements, to form a packed value. Then this value is encrypted as group(b). Here, the extra bit is to ensure that the calculation in the next step will be accurately performed.

2. During query phase, for a feature descriptor, all the vectors are packed in the same way, while the extra bit is now “1”. This value is denoted as group(a). It will always be a positive value.

3. Now the intermediate result is group(a) − group(b). It is known as difference.

4. After Server 2 decrypts the intermediate result, it will be able to recover the value by some simple operation. The intermediate result will be first separated into 64 segments, 26 bits for each. Each of them contains the extra bit following by a 25-bit value α. By examining the extra bit, the absolute value of the difference can be recovered as shown in (6).

\[
\text{difference} = \begin{cases} \alpha, & \text{if the extra bit } = 1 \\ 2^{26} - \alpha, & \text{if the extra bit } = 0 \end{cases}
\]  

After these steps, Server 2 can proceed to next calculation. Experiments of this scheme have been done using C++ and SeComLib library. The results show that the calculation can be correctly performed using a 2048-bit key. In this way, the ciphertext size for every feature descriptor is reduced to 1/64 of the original. Thus, it is possible to store and transfer less data with this scheme.

5. Conclusion

In this paper, we proposed a privacy-enhanced scheme for image similarity search in cloud image database. Inspired by the idea of searchable encryption, this scheme takes advantage of existing feature extraction techniques and use feature descriptors as index for each image. The index, as well as the corresponding image, is encrypted before outsourced to the cloud. The Paillier cryptosystem and a secure multi-party computation scheme are used to ensure functionality and security of the system. Although we introduced a performance improvement method, the search process is linear. Our future work includes reducing the time complexity and further analyzing the security of the system. Experiments should also be done to evaluate its performance.

References