International Journal of Security and Its Applications
Vol.9, No.8 (2015), pp.1-14
http://dx.doi.org/10.14257/ijsia.2015.9.8.01

Secure Similarity Search over Encrypted Cloud Images

Yi Zhu[1,2], Xingming Sun[1,2], Zhihua Xia[1,2] and Naixue Xiong[3]

[1]*Jiangsu Engineering Center of Network Monitoring*
[2]*School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, China*
[3]*School of Computer Science, Colorado Technical University, USA*
zhuyi1344@gmail.com; sunnudt@163.com; xia_zhihua@163.com; xiongnaixue@gmail.com

Abstract

With the growing popularity of cloud computing, more and more data owners are willing to outsource their data to the cloud. However, private data should be encrypted before outsourcing for security requirements, which obsoletes data utilization like content-based image retrieval. In this paper, we propose a secure similarity image search scheme, which allows data owners to outsource their encrypted image database to the cloud server without revealing the real content of images. The proposed scheme supports both global and local feature based image retrieval under various distance metrics, such as earth mover's distance. Firstly, the data owner extracts either global features or local features from images to represent the images. Then, these features are used to generate a searchable index. Finally, both image database and searchable index are encrypted before outsourcing to the cloud server. When a query image coming, the data user extracts feature from the query image and generates the search trapdoor. The trapdoor is then sent to the cloud server and used to compare the similarity with the searchable index. Extensive experiments are conducted to show the efficiency and applicability of our proposed similarity image search system.

Keywords: *similarity image retrieval, linear programming, earth mover's distance, cloud computing*

1. Introduction

Due to the low-cost storage, the number of images is growing rapidly and images are starting to play an important role in our daily life. The storage and retrieval of large-scale image databases has become a big problem. Fortunately, content-based image retrieval is helpful to real-world image retrieval applications. For example, doctors may retrieve the similar cases of their patients in the medical image database to help them make the right decision. However, millions of images are usually included in a large-scale image database and every image is high-dimensional. So, this kind of image retrieval service usually has high computational complexity and intensive storage requirement.

Thanks to the strong data storage and management ability of cloud computing, data owners are willing to store their data on the cloud server without maintaining the image database locally. Authorized data users can retrieve similar images from the cloud server without interacting with the data owner. Despite of various advantages of cloud services, the privacy of image database becomes the main concern of image database outsourcing. Patients do not hope to reveal any information of their medical images to unauthorized user. We study the whole problem thoroughly and propose a practical similar search solution over encrypted images which protect the privacy of image database.

In this paper, we study the secure similarity image search problem and propose a practical solution. We exploit techniques from security, image processing and information

ISSN: 1738-9976 IJSIA

retrieval domains to achieve secure and efficient searching over encrypted images. The solution supports both global and local features under different distance metrics, such as Euclidean distance and earth mover's distance based methods. In particular, a secure linear programming (LP) transformation is designed such that the cloud server is able to determine the earth mover's distance with input and output privacy. By leveraging the computation power of the cloud, the proposed scheme costs low local computation while achieving high retrieval accuracy. The extensive experiments are conducted to validate the efficiency and applicability of the proposed solution.

The reminder of this paper is organized as follows. Section introduces the system architecture and preliminaries. In Section, we give the design of similarity image search, which supports both global and local feature based solution. In Section, we formally analyze the security of the proposed schemes. In Section, we implement our proposed scheme and study its efficiency, applicability and local computational savings. Finally, Section summarizes related work, and a conclusion is given in Section.

2. System Overview and Preliminaries

Figure 1. System Architecture

2.1. System Architecture

As shown in Figure 1, the proposed similarity image search system involves three types of entities: data owner, data user and cloud server.

The data owner has a large-scale of image database to be outsourced, where n is the number of images in the database. The data owner extracts features from images and generates a searchable index. To protect the sensitive information of images, the image database and searchable index are encrypted before outsourcing to the cloud server. To support the similarity image search over encrypted images, the data owner has to construct a searchable encryption scheme.

The authorized data user extracts feature vector from the query image and generates an encrypted query. Then, the encrypted query is submitted to the cloud server.

The cloud server receives the encrypted query and compares the similarities with the searchable index. Then, the top-ranked similar encrypted images are returned to the data user. Finally, the data user decrypts the received images.

The data owner and data user are always trusted, but the cloud server is considered to be “honest but curious”. The cloud server tries to derive sensitive information by analyzing the communication history. Our proposed similarity image search solution is designed to prevent the cloud server from knowing either the image database or users' queries.

2.2. Earth Mover’s Distance

The earth mover's distance [1-2] compares the similarity between the distributions. Given two distributions, the distribution with smaller sum of weights can be viewed as a mass of earth which rightly spread in space, the distribution with larger sum of weights

can be viewed as an array of holes in the same space. The EMD measures the minimal cost of moving all the earth into holes. Transporting a unit earth for a unit distance defines a unit of work. The EMD transforms the matching problem to the transportation problem, two distributions have the least transportation cost can be viewed as the most similar ones. Given two image signatures $S_t = \{(s_1^{(t)}, w_1^{(t)}), (s_2^{(t)}, w_2^{(t)}), ..., (s_{m_t}^{(t)}, w_{m_t}^{(t)})\}$ for t=1,2, the EMD is defined in terms of an optimal flow $F = \{f_{i,j}\}$, which minimizes the work required to move earth from one signature to another, donated as $W(S_1, S_2, F) = \sum_{i=1}^{m_1}\sum_{j=1}^{m_2} f_{i,j} d_{i,j}$, where $d_{i,j} = d(s_i^{(1)}, s_j^{(2)})$ is the ground distance between $s_i^{(1)}$ and $s_j^{(2)}$, e.g. the Euclidean distance in $\Re^d$. The flow $f_{i,j}$ must satisfy the following constraints:

$$f_{i,j} \geq 0, 1 \leq i \leq m_1, 1 \leq j \leq m_2;$$
$$\sum_{i=1}^{m_1} f_{i,j} \leq w_j^{(2)}, 1 \leq j \leq m_2;$$
$$\sum_{j=1}^{m_2} f_{i,j} \leq w_i^{(1)}, 1 \leq i \leq m_1;$$
$$\sum_{i=1}^{m_1}\sum_{j=1}^{m_2} f_{i,j} = \min(\sum_{i=1}^{m_1} w_i^{(1)}, \sum_{j=1}^{m_2} w_j^{(2)}).$$

Once the optimal flow $f_{i,j}^*$ is found, the EMD between S_1, S_2 is defined as

$$EMD(S_1, S_2) = \frac{\sum_{i=1}^{m_1}\sum_{j=1}^{m_2} f_{i,j}^* d_{i,j}}{\sum_{i=1}^{m_1}\sum_{j=1}^{m_2} f_{i,j}^*} \quad (1)$$

where numerator is the minimal transportation cost and the denominator is the total movements. The signatures can always be converted to proper probability distributions by normalizing the weights to add up to 1 [3]. Hence, we can simply the constraints to:

$$f_{i,j} \geq 0, 1 \leq i \leq m_1, 1 \leq j \leq m_2;$$
$$\sum_{i=1}^{m_1} f_{i,j} = w_j^{(2)}, 1 \leq j \leq m_2;$$
$$\sum_{j=1}^{m_2} f_{i,j} = w_i^{(1)}, 1 \leq i \leq m_1;$$
$$\sum_{i=1}^{m_1}\sum_{j=1}^{m_2} f_{i,j} = \sum_{i=1}^{m_1} w_i^{(1)} = \sum_{j=1}^{m_2} w_j^{(2)} = 1.$$

3. The Proposed Scheme

In this section, we describe the design of similarity image search system for both global and local features.

3.1. Notations

- M, S, ξ: image database, signature database, centroid database.
- K_M, K_S, K_ξ: secret key for image, signature, centroid database encryption.
- $K_j, H_j, j = 1, ..., \lambda$: secret key, LSH functions for index construction.
- S_q: signature of query image.
- ϕ: one way hash function.
- I, T_{ξ_q}: searchable index, trapdoor.
- $R_{S_l'}$: retrieval encrypted signature set.
- $R_{\varepsilon(M_l)}$: top-k ranked order encrypted image set.
- Ω: transformed EMD optimization problem set.

3.2. Global Feature based Scheme

For global feature based scheme, one global feature vector $\boldsymbol{g}_i$ is extracted from each image, and Euclidean distance is the most commonly used metric. Naively, we have to compute the distances between the query vector $\boldsymbol{g}_q$ and each global feature vector $\boldsymbol{g}_i$ in the database one by one. To improve efficiency, we propose a two-stage scheme: in the first stage, the server quickly filters out those images that are very unlikely to be in the top list; in the second stage, the server then performs accurate distance comparison over the refined image subset.

For the first stage, we need to build a pre-filter table. We utilize the well-known locality sensitive hashing (LSH) [4-7] to cluster the image database such that similar images are hashed together so as to reduce the searching complexity. A LSH family is called (t, ct, p, q)- sensitive if

$$Pr(h(x) = h(y)) \geq p \text{ for } d(x, y) \leq t$$
$$Pr(h(x) = h(y)) \leq q \text{ for } d(x, y) \geq ct$$

where the probabilities $p > q$ and the constant $c > 1$. As using one hash function instance $h(\square)$ may not give good enough results for the locality sensitive hash, in practice, it is suggested to use a set of e independent hash functions $\{h_1, ..., h_e\}$, and the final hash digest is obtained by concatenating their outputs, denoted as $\mathrm{H}(\mathbf{x}) = \{h_1(\mathbf{x}), ..., h_e(\mathbf{x})\}$, which maps an ℓ-dimensional vector $\mathbf{x}$ to a e-dimensional one. Clearly, increasing e values would enlarge the gap between the probabilities of collision for close points p^e and far points q^e.

In particular, the data owner randomly chooses the e LSH functions and applies to the centroid database so as to build a key-value based hash table. Let $\{w_i\}_{i=1}^{N}$ denote the derived set of LSH hash digests, where N refers to the total number of clusters. To increase the clustering accuracy, we can repeat this process λ times by generating λ hash tables. Indeed, the parameters e and λ can be adjusted such that it performs favourably to maintain both low false negatives, as suggested by the methodology from LSH community [8].

Table 1. An Example of j-th Hash Table

$\phi(k_j, wd_{1,j})$	$ID(m_7), ID(m_9), ID(m_{12}), ID(m_{17}), \cdots$
$\phi(k_j, wd_{2,j})$	$ID(m_{13}), ID(m_{24}), ID(m_{35}), ID(m_{67}), \cdots$
$\phi(k_j, wd_{3,j})$	$ID(m_{27}), ID(m_{49}), ID(m_{62}), ID(m_{73}), \cdots$
……	……
$\phi(k_j, wd_{\mathrm{N}_j,j})$	$ID(m_{47}), ID(m_{59}), ID(m_{92}), ID(m_{117}), \cdots$

However, in general, such LSH is not necessary to have one-way property. Therefore, we cannot directly outsource this pre-filter table to the cloud server as it may leak information about the centroid database to the adversary. To enhance the security, a pseudo-random permutation (PRP) $\phi: \{0,1\}^{\ell_h} \times \kappa \to \{0,1\}^{\ell_h}$ is applied to protect the keywords in the table, as shown in Tab. 1. The formal description is elaborated in Tab. 2.

Table 2. Building Pre-filter Table

Data owner chooses λ independent LSH functions $\{H_j\}_{j=1}^{\lambda}$ by setting $H_j = (h_{1,j}, h_{2,j}, ..., h_{e,j})$,where $h_{1,j}, h_{2,j}, ..., h_{e,j}$ are randomly chosen from the LSH family H

For each $j = 1,...,\lambda$, data owner builds the *j*-th hash table by applying function H_j over all the centroid database ξ. One example is shown in Table 1.
For each $j = 1,...,\lambda$, data owner picks a random key k_j and replaces each LSH hash digest $w_{i,j}, i = 1,...,N_j$ in the *i*-th hash table with $\phi(k_j, w_{i,j})$.
For each $j = 1,...,\lambda$, data owner further fills the *j*-th hash tables with identifiers of corresponding images $ID(m_i)$.

After building the pre-filter table, the data owner outsources it together with the rest secure searchable index to the cloud server. To search the image database, a user first extracts the query feature vector $\boldsymbol{g}_q$ from the query image. He/She then generates $\phi(k_j, H_j(\boldsymbol{g}_q))$ for $j = 1,...,\lambda$ as part of his search query Q to the cloud. For each $j = 1,...,\lambda$, the cloud server first retrieves the identifiers from the buckets $\phi(k_j, H_j(\boldsymbol{g}_q))$ in the *j*-th hash table, and then performs the secure distance comparison over the encrypted features, which is designed as follows.

After narrowing down the search scope, we want to directly compare the underlying distance metrics to retrieve more accurate results. We have to encrypt the global features in such way that the cloud server is able to compare the similarity. For each image feature vector $\boldsymbol{g}_i = (g_{1,i},...,g_{\ell,i})^T$ to be outsourced in the cloud, the data owner first constructs an modified vector as $\boldsymbol{g}_i = (g_{1,i},...,g_{\ell,i},\| \boldsymbol{g}_i \|_2^2)^T$. Next, he randomly picks a $(\ell+1)\times(\ell+1)$ invertible matrix $\boldsymbol{R}$ to encrypt the extended feature vector as $\boldsymbol{g}_i^{'} = \boldsymbol{R} \cdot \boldsymbol{g}_i$. For a query feature vector $\boldsymbol{g}_q = (g_{1,\mathrm{q}},...,g_{\ell,\mathrm{q}})^T$, the data user first constructs an modified vector as $\boldsymbol{g}_q = (-2g_{1,\mathrm{q}},...,-2g_{\ell,\mathrm{q}},1)^T$. He next chooses a random positive value *r* and uses it with the secret matrix $\boldsymbol{R}$ to encrypt the modified query vector as $\boldsymbol{g}_q^{'} = r\boldsymbol{R}^{-1} \cdot \boldsymbol{g}_q$. Upon receiving the encrypted query feature vector, the cloud conducts the scalar product

$$\boldsymbol{g}_q^{'T} \cdot \boldsymbol{g}_i^{'} = (r\boldsymbol{R}^{-1} \cdot \boldsymbol{g}_q)^T \cdot \boldsymbol{R}^T \boldsymbol{g}_i = r\boldsymbol{g}_q^{\ T} \cdot \boldsymbol{g}_i = r(\| \boldsymbol{g}_i \|_2^2 - 2\sum_{j=1}^{\ell} g_{j,i} g_{j,q}) = r(\left\| \boldsymbol{g}_q - \boldsymbol{g}_i \right\|_2^2 - \left\| \boldsymbol{g}_q \right\|_2^2)$$

Here the distance $\| \boldsymbol{g}_q - \boldsymbol{g}_i \|_2^2$ is hidden by the secret scalar r and the unknown $\| \boldsymbol{g}_q \|_2^2$. We note that, when $x \geq 0$, the function $f(x) = x^2$ is order preserving, i.e., $f(x_1) > f(x_2)$ implies $x_1 > x_2$. Also, because for each query $\boldsymbol{g}_q$ the values of $\left\| \boldsymbol{g}_q \right\|_2^2$ and r is fixed, the cloud server can directly find closest feature vectors by simply sorting out the set of scalar product, without knowing the sensitive information from the feature vectors.

3.3. Local Feature based Scheme

For local feature based scheme, a set of local features are extracted from local regions of an image. One popular approach of content based image retrieval using local features is called bag-of-words model. In this model, local features are extracted from all images in the database and jointly clustered. The cluster centers are used as cluster identifiers, which form the vocabulary. After clustering, only the identifier of the most similar cluster center is kept for each local feature, which signifies a word. Then the cluster occurrence histogram is created for each image, which can be represented as a vector of occurrence counts of the local features. Through this way, an image can be expressed as a bag of words in a visual vocabulary.

By comparing the similarity of histograms, one can retrieve similar images. As mentioned before, EMD distance is used during the search process. Denote the signature of each image in the database as $S_t = \{(s_1^{(t)}, w_1^{(t)}), (s_2^{(t)}, w_2^{(t)}), ..., (s_{m_t}^{(t)}, w_{m_t}^{(t)})\}$ for

$t=[n]$. Let $S_q=\{(s_1^{(q)},w_1^{(q)}),(s_2^{(q)},w_2^{(q)}),...,(s_{m_q}^{(q)},w_{m_q}^{(q)})\}$ be the signature of a query image. The EMD distance can be converted to an LP optimization problem as follows:

$$\begin{aligned}\text{minimize}\quad & \sum_{i=1}^{m_t}\sum_{j=1}^{m_q} f_{i,j}d_{i,j}\\ & f_{i,j}\geq 0, 1\leq i\leq \mathrm{m}_t, 1\leq j\leq \mathrm{m}_q;\\ & \sum_{i=1}^{m_t} f_{i,j}=w_j^{(q)}, 1\leq j\leq \mathrm{m}_q;\\ \text{subject to}\quad & \sum_{j=1}^{m_q} f_{i,j}=w_i^{(t)}, 1\leq i\leq \mathrm{m}_t;\\ & \sum_{i=1}^{m_t}\sum_{j=1}^{m_q} f_{i,j}=1.\end{aligned}$$

Similarly, we first need to build a pre-filter table for sub-linear searching time. In the EMD case, we need to filter the images using a quick estimation of EMD without computing it. We utilize an easy-to-compute EMD lower bound as the estimation. The Euclidean distance between the centroids of two signatures with the same total weights is a lower bound on the EMD between them. It is well known that [1]

$$EMD(S_t,S_q)=\sum_{i=1}^{m_t}\sum_{j=1}^{m_q} f_{i,j}{}^{*}d_{i,j}\geq\left\|\sum_{i=1}^{m_t}s_i^{(t)}w_i^{(t)}-\sum_{j=1}^{m_q}s_j^{(q)}w_j^{(q)}\right\|_2$$

Similar to the global feature case, we apply one-way LSH functions on centroid database ξ, then groups the signature *ID* set on all λ hash tables. After that, the data owner outsources the secure searchable index, encrypted signature database and encrypted image database to the cloud server. When image query comes, the data user first generates the signature $\boldsymbol{Q}$, then compute the centroid ξ_q. After that, he applies λ LSH functions and one-way function to ξ_q to compute the trapdoor $T_{\xi_q}=\phi(K_j,H_j(\xi_q)), j=1,...,\lambda$ and sends it to the cloud server. The cloud server retrieves the corresponding encrypted signatures and sends them to the data user. By building the secure searchable index, we can achieve the sub-linear filtering complexity for the large-scale image database. In the second stage, we will directly compute the underlying distance metric, which is EMD, to compare the similarities of different images. We will explain in the next section.

3.4. LP Transformation

The local features always involve in the more complex distance metrics, which need more delicate security design. In the second stage, after receiving the encrypted signatures from the cloud in the first stage, the data user decrypts the signatures and computes the distance c_{ij} between the cluster $\boldsymbol{s}_i$、$\boldsymbol{q}_j$. Then he formulates the EMD optimization problem as in Eq. (2). The data user wants to leverage the computation power of cloud server to compute EMD with privacy protection. Next, we will explain how to make the cloud server securely compare the EMD distance between different images without revealing the sensitive information. For more concise explanation, we use the matrix expression for Eq.(2):

$$\begin{aligned}\text{minimize}\quad & \mathbf{c^T x}\\ \text{subject to}\quad & \mathbf{Ux}=\tau\\ & \mathbf{Vx}\leq\mathbf{E}\end{aligned}\qquad(2)$$

where $\boldsymbol{c}$ is an $ab\times 1$ distance vector, $\mathbf{x}$ is an $ab\times 1$ flow vector. $\mathbf{U}$ is an $1\times ab$ known structure matrix and $\mathbf{V}$ is an $(a+b)\times ab$ known structure matrix. τ is the minimal total

weight and $\boldsymbol{E} = (w_{s_i}, w_{q_j})$ is the weight vector, where $1 \le i \le a, 1 \le j \le b$. We denote the problem (2) by a tuple $\Psi = (\mathbf{c}, \mathbf{U}, \tau, \mathbf{V}, \mathbf{E})$. Our design goal is to find a secure and efficient transformation, using a secret key K_T, to make cloud compute the EMD optimization problem while protecting the input and output privacy. It means that the cloud solves the randomly transformed EMD optimization problem without knowing the input distance vector $\boldsymbol{c}$, weight vector $\mathbf{E}$ and the output flow vector $\mathbf{x}$. After that, we also want cloud to compute the EMD distance and sort the results. Our secure transformation contains two main steps. First of all, to protect the output privacy, we perform the affine mapping that $\mathbf{x} = \Lambda\mathbf{y} - \gamma$, where Λ is an $ab \times ab$ non-singular matrix and γ is an $ab \times 1$ vector. The original problem is transformed to:

$$\begin{aligned} &\text{minimize } \mathbf{c}^T\Lambda\mathbf{y} - \mathbf{c}^T\gamma \\ &\text{subject to } \mathbf{U}\Lambda\mathbf{y} = \tau + \mathbf{U}\gamma, \\ &\qquad \mathbf{V}\Lambda\mathbf{y} \le \mathbf{E} + \mathbb{V}\gamma \end{aligned} \tag{3}$$

Next, we multiply the $((a+b)\times(a+b))$ generalizes permutation $\mathbf{G}$ to the inequality constraints to protect $\mathbf{E}$ and multiply a real positive value *r* to protect optimal value. We transform the problem (3) to another problem:

$$\begin{aligned} &\text{minimize } r\mathbf{c}^T\Lambda\mathbf{y} - r\mathbf{c}^T\gamma \\ &\text{subject to } \mathbf{U}\Lambda\mathbf{y} = \tau + \mathbf{U}\gamma, \\ &\qquad \mathbf{G}\mathbf{V}\Lambda\mathbf{y} \le \mathbf{G}(\mathbf{E} + \mathbb{V}\gamma) \end{aligned} \tag{4}$$

Because the constant term $r\mathbf{c}^T\gamma$ does not affect the optimal solution, the final transformation problem can be formed as:

$$\begin{aligned} &\text{minimize } \mathbf{c}'^{\boldsymbol{T}}\mathbf{y} \\ &\text{subject to } \mathbf{U}'\mathbf{y} = \tau', \\ &\qquad \mathbf{V}'\mathbf{y} \le \mathbf{E}' \end{aligned} \tag{5}$$

where $\mathbf{c}'^{\boldsymbol{T}} = r\mathbf{c}^T\Lambda$, $\mathbf{U}' = \mathbf{U}\Lambda$, $\tau' = \tau + \mathbf{U}\gamma$, $\mathbf{V}' = \mathbf{G}\mathbf{V}\Lambda$, $\mathbf{E}' = \mathbf{G}(\mathbf{E} + \mathbf{V}\gamma)$. This problem has similar structure to the original problem (2). We use $\Psi_{K_T} = (\mathbf{c}', \mathbf{U}', \tau', \mathbf{V}', \mathbf{E}')$ to denote the secure transformed problem, where the secret transformation key $K_T = (\mathbf{G}, \Lambda, \gamma, r)$. The whole transformation process is illustrate in Algorithm 1. After solving the EMD optimization problem, we also want cloud to compute EMD as in Eq. (1) and sort the results. Observe that the numerator of EMD equation is the optimal values of the original problem (2) and the denominator is the sum of the elements of optimal solution. However, the cloud server solves the transformed problem (5) instead of the original one for security reason. The difference of optimal value between original problem and transformed problem is a constant term $r\mathbf{c}^T\gamma$, which can be computed by data users before outsourcing. We also want the sum of elements of optimal solutions between original problem transformed problem differs in a constant term after the affine mapping $\mathbf{x} = \Lambda\mathbf{y} - \gamma$. So we apply one additional constraints when constructing Λ such that for Λ^{-1} each column's sum is one. Then we can derive that $\sum_{i=1}^{a}\sum_{j=1}^{b} x_{ij} = \sum_{i=1}^{a}\sum_{j=1}^{b} y_{ij} - \sum_{k=1}^{ab}\gamma_k$. The data user can compute $\sum_{k=1}^{ab}\gamma_k$ before outsourcing. When outsourcing the secure transformed problems to cloud server, the data users also send two offset constants $r\mathbf{c}^T\gamma$ and $\sum_{k=1}^{ab}\gamma_k$ to cloud server. Then the cloud server can solve the transformed problems and use these constants to compute the right EMD in an order preserving way. The offset constants themselves will reveal little information. After that, the cloud server sorts the results and returns the more accurate ranked order encrypted images.

3.5. Construction Detail

In this section, we will describe the total service flow of secure similarity image search scheme.

- **KenGen**(K_s) The data owner generates secret key $\mathcal{K}$, which contains $K_M, K_S, K_\xi, K_j, j = 1,...,\lambda$, , using master key K_S .
- **BuildIndex**(ξ, H_J, K_J, ϕ) The data owner builds the secure LSH index for the centroids database, using LSH functions H_J , keys K_J and one-way hash function ϕ .
- **DataEnc**(M, S, K_M, K_s) The data owner encrypts the image database M using key K_M to form the encrypted database ε_M . He also encrypts the signature database S using key K_S to form the encrypted signature database ε_S . We can use the RSA or AES ciphers to encrypt them. The data owner sends the encrypted image database, encrypted signature database and the secure index to cloud server. The authorized data users can retrieve the ranked order encrypted outsourced images later from the cloud server. To achieve this functionality, the data owner needs to sharing the following information with data users: $K_M, K_S, K_\xi, K_j, j = 1,...,\lambda$. The data user receives these shared information to preprocess the image query.
- **TrapdoorGen**(S_q, H_J, K_J, ϕ) For a given image query, the data user first extracts its signature $\mathbf{S_q}$ using the clustering algorithm, then computes its centroid ξ_q . After that, the data owner applies LSH functions H_j , secret keys K_j , $j = 1,...,\lambda$ and one way hash function ϕ to construct trapdoor $T_{\xi_q} = (\phi(K_1, H_1(\xi_q)),...,\phi(K_\lambda, H_\lambda(\xi_q)))$ .Then the data user sends the trapdoor T_{ξ_q} to the cloud server.
- **SearchIndex**(I, T_{ξ_q}) After receiving the trapdoor T_{ξ_q} , the cloud server uses subtrapdoor $T_{\xi_q, j} = \phi(\mathrm{K}_j, H_j(\xi_q))$ to identify the relevant bucket whose identifier has the same value as the subtrapdoor. The cloud server retrieves all the buckets in the λ hash tables and forms the distinctive centroid *ID* set. Next, it sends the corresponding encrypted signature set to the data user.
- **SecureTrans**($R_{s_i'}, S_q, K_S$) After receiving the encrypted signature set, the data user first decrypts them using secret key K_S . Then, the data user computes the distance vector $\boldsymbol{c}_{\mathbf{lq}}{}^{T}$ for each retrieved signature $\boldsymbol{S_l}$ and querying signature $\boldsymbol{S_q}$. He also uses the weights information to form the EMD optimization problem for each retrieved and querying signature pair. After that, the data user formulates the EMD optimization problem and generates secure transformation key $K_{T_l} = (\mathbf{G_1}, \Lambda_1, \gamma, r)$. Then he transforms the original problem in Eq. (2) to Eq. (5) and computes two offset constants $r_l \mathbf{c_{lq}}^{T} \gamma_l$ and $\sum_{k=1}^{ab} \gamma_{kl}$. Next, the data user outsources all the secure transformed EMD optimization problem set and corresponding offset constants to cloud server. Note that for each retrieved and querying signature pair, we adopts different transformed key K_{T_l} , which can protect the sensitive information in a one-time pad manner.
- **DistanceCom** (Ω) After receiving the secure transformed EMD optimization problem set and the corresponding offset constants, the cloud server solves the LP problem using the existing solving algorithm without learning the sensitive information. After that, it uses the offset constants to compute the EMD in an order preserving way. Next, the cloud server sorts the EMD distance and returns the ranked order encrypted images to data user.

- **DataDec**($R_{\varepsilon(\mathbf{M})}, K_M$) After retrieving the ranked order encrypted images, the data user decrypts data using the secret key K_M and gets the original similar images according to the search request.

3.6. Global Feature vs. Local Feature

Under our two stages similarity image search system, the global and local feature based solutions have certain differences. The global feature based solution only needs one round communication between data user and cloud server to retrieve the ranked order encrypted images, while local feature based solution needs two rounds. There are two reasons. The first reason is that the underlying distance metric for local feature based solution, which is EMD, has more complicated problem structure and needs more delicate security design. Our encryption method should preserve the LP problem's structure and make sure the cloud server can solve the right optimal solution. We need to compute the distance vector $\boldsymbol{c}_{zq}^T$ for each retrieved and querying signature pair first, then encrypt this distance vector using transformation key Λ, r. We cannot outsource these two operations to cloud server at the same time without revealing the distance vector and the transformation key. The second reason is that their is a trade-off between communication round and security. For one round solution, every feature vector encrypted use the same invertible matrix W. It is vulnerable to know plaintext attack, when cloud server have certain number of pairs of plaintext and ciphertext feature vectors, it can derive the invertible matrix W. For two rounds solution, the data owner adopts the standard encryption techniques, which can resist the known plaintext attack. So we favor the two rounds design for our system, which also has higher scalability for different traditional and underlying distance metrics.

4. Security Analysis

It is well known that the server's computational complexity is at least $O(n)$ if we want to achieve perfect query privacy. In order to have efficient (sub-linear) system, it is necessary to reveal minimum database information to the server. For both global and local feature based scheme, we use pre-filter table to group similar images together. Therefore, the adversary knows those images in the same bucket are similar to each other. In addition, the server also knows those images selected by the same query are similar to each other as well. The above information leakage is a compromise for efficiency.

Now we are going to argue that the encrypted database, secure searchable index and encrypted query does not reveal extra information to the adversary. First of all, it is easy to see that the image database is protected if the encryption scheme is CPA-secure. The keywords in hash tables are encrypted by deterministic PRP, and the same keywords are never encrypted with the same key twice; thus, the encrypted keywords are indistinguishable from random. The global feature vectors $\boldsymbol{g}_i$ are protected by a random invertible matrix $\boldsymbol{R}$, so each is indistinguishable from a random vector. The same argument holds for the query vector $\boldsymbol{R}^{-1}\boldsymbol{g}_q$.

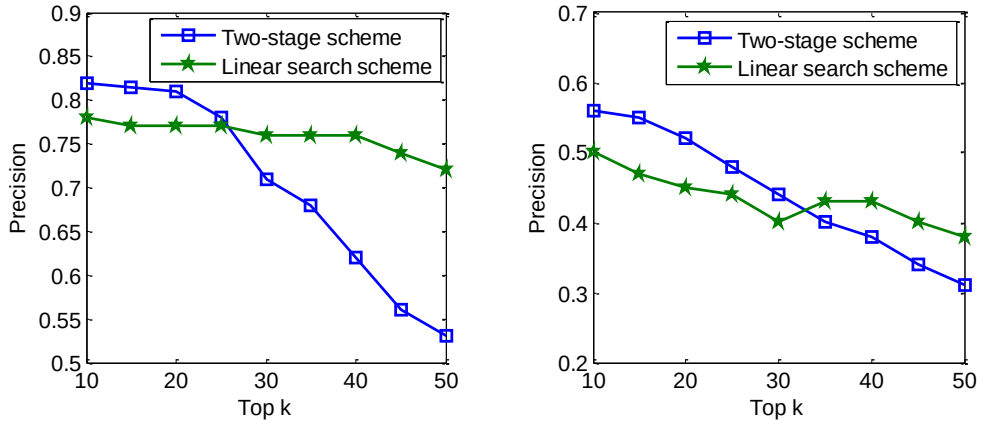
In terms of the local feature based scheme, the main concern is the LP transformation for EMD distance. In the LP transformation, to protect the output privacy, we firstly perform the affine mapping that $\mathbf{x} = \Lambda\mathbf{y} - \gamma$, where Λ is an $ab \times ab$ non-singular matrix and γ is an $ab \times 1$ vector. Secondly, we multiply the $((a+b)\times(a+b))$ generalizes permutation $\mathbf{G}$ to the inequality constraints to protect $\mathbf{E}$. Through above two steps, the original LP problem is transformed and the privacy of original problem is well protected. From the cloud's view, the transformed LP problem is indistinguishable from the original one. Therefore, the security of local feature based scheme is guaranteed by the LP transformation.

5. Implementation and Performance

In this section, we demonstrate a thorough experimental evaluation of the proposed similar image search design on a real-world image database: Corel test set. The whole experiment system is implemented by C++ language on a Windows Server with Intel Core2 Processor 2.0GHz. The performance of our technique is evaluated regarding the efficiency of the proposed scheme, as well as the search precision.

5.1. Precision

In this paper, we propose both global and local feature based similar image search schemes. The proposed schemes have two stages. In the first stage, the cloud server filters out irrelevant images using pre-filter tables. In the second stage, the cloud server performs accurate distance comparison over the refined image subset. In order to show that the proposed scheme does not decrease the search accuracy, we compare the search accuracy of global and local feature based schemes to linear search scheme. We define a measure as $P_k = k'/k$ where k' is the number of real top-images that are returned by the cloud server.



(a) Precision of Global feature scheme

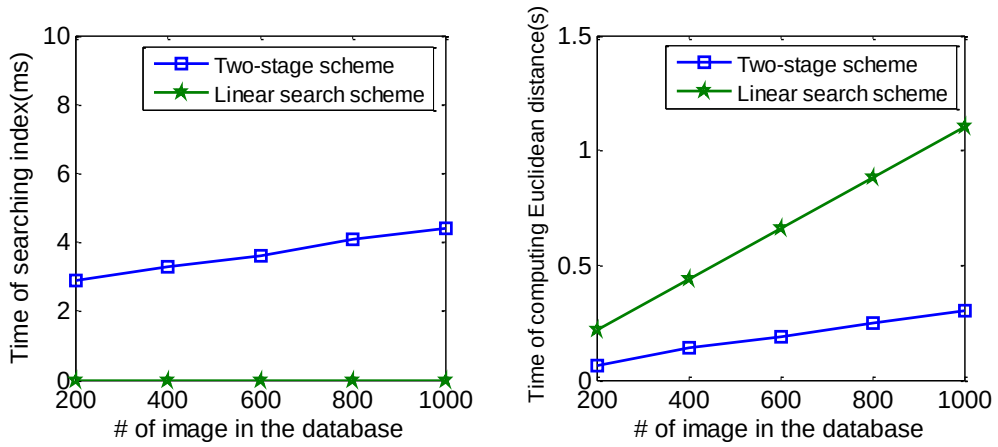
(b) Precision of Local feature scheme

Figure 2. Search Precision

Figure 2(a) is the search precision of global feature based scheme. We can learn from Figure 2(a) that when, two-stage scheme achieves higher search precision than linear search scheme. Figure 2(b) is the search precision of local feature based scheme. When $k \le 30$, the search precision of two stage scheme is better than the linear search scheme. From Fig. 2, we can see that two stage scheme does not decrease the search accuracy, and even achieves higher search precision than linear search scheme for certain *k*. In our experiments, we do not compare the precision of global feature based scheme to that of local feature based scheme. Our purpose is to show that two stage scheme is applicable to both global and local feature based image retrieval. The experiments also give the same conclusion that LSH algorithm is applicable to pre-filter irrelevant images in content based image retrieval.

5.2. Efficiency

1) Query: Query execution at the cloud server side consists of searching index, computing and ranking the accurate distance of similar images. Since the linear search scheme does not build a search index, so it is obvious to know that the search overhead of linear search scheme is linear to the number of image. Figure 3 shows the time cost of searching index and computing Euclidean distance of global feature based scheme. We can see from Figure 3(a) is the comparison of searching index time. The linear search scheme does not build index, so the search index time is always 0. Figure 3(b) shows the time cost of computing and ranking Euclidean distance. We can see that the time cost of two stage scheme increases slowly while the time cost of linear search scheme is linear to the size of image database.

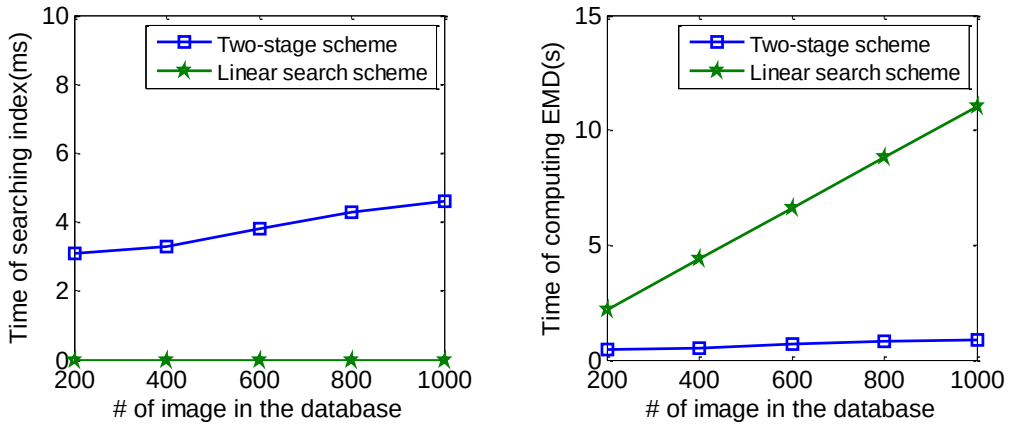


(a) Time of Searching Index **(b) Time of Computing Euclidean Distance**

Figure 3. Search Efficiency of Global Feature Scheme

Figure 4 shows the time cost of searching index and computing EMD distance of local feature based scheme. The search time of two stage scheme slowly grow with the increase number of image. For the same reason as global scheme, the time of searching index for linear search is still 0. Figure 4(b) is the time cost of computing EMD distance. The computation of EMD distance is complicated, so the time overhead is much larger (almost 10 times) than Euclidean distance.

Figure 3 and Figure 4 show us that the time cost of computing accurate distance is much larger(almost 100 times) than the time of searching index. Comparing to the time of computing accurate distance, the time of searching index is negligible. Thus, the total time cost of searching similar images can be represented by the time cost of computing accurate distance. Besides, because of complicated calculation process, the local feature based scheme costs more time than global feature based scheme on calculating accurate distance. The proposed two stage scheme first filters irrelevant images and narrow down the scope of images that need to directly compute accurate distance. However, linear search scheme needs to compute the accurate distance between query image and all images in the database. So, the two stage scheme has better search efficiency than linear search scheme. By analyzing the accuracy and efficiency of two stage scheme, we can conclude that locality sensitive hashing algorithm is applicable to the similarity search of encrypted images.



(a) Time of Searching Index **(b) Time of Computing EMD Distance**

Figure 4. Search Efficiency of Local Feature Scheme

6. Related Work

Searchable symmetric encryption (SSE) on text domain has been widely studied in the literature. Song *et al.* [9] first proposed the notion of searchable encryption. They design a special two-layered encryption scheme to reduce the round complexity. They use the symmetric key setting and the searching overhead is linear to the length of total file collection length. Goh *et al.* [10] defines the secure index which possesses the characteristics of semantically security against adaptive chosen keyword attack (IND-CKA). They use the Bloom filters and pseudo-random functions to construct the secure index, for which the searching complexity is proportional to the number of files contained this keyword in the collection. Curtmola *et al.* [11] propose a keyword based scheme, in which an encrypted hash table index is built for each single file. This scheme is very efficient and there is constant search complexity for each returned file. It also can achieve adaptive SSE security. Li *et al.* [12] for the first proposed fuzzy keyword search over encrypted cloud data in cloud computing. They exploit edit distance to quantify keyword similarity. Fuzzy keywords tolerates errors to some extent, it is only applicable to strings under edit distance. If we have long words, then the fuzzy keywords set is very big which is inefficient for search. Kuzu *et al.* [13] proposed an scheme for similarity search over encrypted data. They utilize locality sensitive hashing for fast near neighbor search. Their index is constructed based on LSH algorithm. Their search scheme supports similarity search over encrypted data. Wang *et al.* [14] proposed a ranked keyword search scheme for file retrieval. They use inverted index, ranking function, one-to-many order-preserving mapping to securely return the ranked results and enhance system usability. These works mainly focus on keyword search in text domain, which does not suit CBIR well. Because image has much higher dimensionality compared to keyword, the index size will become too large and make it very inefficient if we apply the same techniques to CBIR. In addition, the images are more expressive and possess more flexibilities, which cannot be limited to the existing distinct keyword set. There are also many decent works focus on image encryption. Chen *et al.* [15] propose a 3D chaotic cat map based symmetric image encryption scheme. This scheme utilizes the properties of bulk data capacity and high redundancy for images and sensitivity to initial conditions for chaotic maps to design a fast and highly secure image encryption, which can resist the statistical and differential attacks. Refregier *et al.* [16] propose a new optical encryption method for images which random encodes in the input and Fourier planes. This encryption scheme transforms the input signal to stationary while noise. We can also use the standard ciphers such as AES to encrypt images. These works mainly focus on image encryption without considering

the search and similarity comparison abilities. Lu *et al.* [17] propose a secure search scheme over encrypted multimedia database. They represent images using visual words and treat them as keywords. Then they adopt the widely used index structures in text domain, which are inverted index and min-Hash, to perform secure search over multimedia data. However, this work does not suit for other image features except visual words. Furthermore, compared with text, image contains more expressive information and flexibilities, the same index will discard more useful information and make the search result less accurate. In short, secure, efficient and accurate search over encrypted images under different features and distance metrics is still an open problem.

7. Conclusion

In this work, we propose a secure similarity image search framework. The proposed framework jointly considers the techniques from security, image processing and information retrieval domains to perform secure and efficient image search over encrypted database, which is an practical effective system to solve the secure image outsourcing problem. The proposed framework supports both global and local features based similar image search over encrypted images. To achieve sub-linear searching time, we design a two-stage structure to take into consideration the trade-off between efficiency and accuracy. In the first stage, the image database is pre-filtered to shrink the search scope. Then in the second stage, those filtered images are compared one by one for refined search results. By leveraging the computation power of the cloud, low local computation is achieved while maintaining high retrieval accuracy, which is suitable for mobile devices in the outsourcing scenario.

Acknowledgements

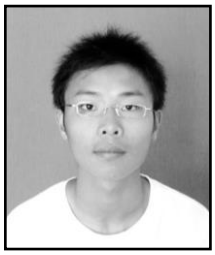
This work is supported by the NSFC (61232016, U1405254, 61173141, 61173142, 61173136, 61373133), 201301030, 2013DFG12860, BC2013012 and PAPD fund.

References

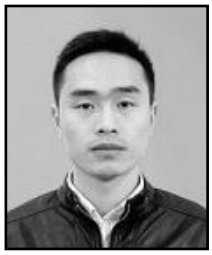
[1] Y. Rubner, C. Tomasi, and L. J. Guibas, “The earth mover’s distance as a metric for image retrieval,” International Journal of Computer Vision, vol. 40, no. 2, **(2000),** pp. 99–121.
[2] H. Ling and K. Okada, “An efficient earth mover’s distance algorithm for robust histogram comparison,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 29, no. 5, **(2007),** pp. 840–853.
[3] E. Levina and P. Bickel, “The earth mover’s distance is the mallows distance: some insights from statistics”, in Computer Vision, 2001, ICCV 2001, Proceedings, Eighth IEEE International Conference on, vol. 2. IEEE, **(2001)**, pp. 251–256.
[4] P. Indyk and R. Motwani, “Approximate nearest neighbors: towards removing the curse of dimensionality”, in Proceedings of the thirtieth annual ACM symposium on Theory of computing, ACM, **(1998)**, pp. 604–613.
[5] A. Gionis, P. Indyk and R. Motwani, “Similarity search in high dimensions via hashing”, in VLDB, vol. 99, **(1999)**, pp. 518–529.
[6] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, “Locality-sensitive hashing scheme based on p-stable distributions”, in Proceedings of the twentieth annual symposium on Computational geometry. ACM, **(2004)**, pp. 253–262.
[7] A. Rajaraman and J. D. Ullman, “Mining of massive datasets”, Cambridge University Press, **(2011)**.
[8] A. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions”, in Foundations of Computer Science, 2006. FOCS’06. 47th Annual IEEE Symposium on. IEEE, **(2006)**, pp. 459–468.
[9] D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data”, in Security and Privacy, 2000, S&P 2000, Proceedings, 2000 IEEE Symposium on. IEEE, **(2000)**, pp. 44–55.
[10] E.-J. Goh, “Secure indexes”, IACR Cryptology ePrint Archive, vol. 2003, **(2003),** p. 216.
[11] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions”, in Proceedings of the 13th ACM conference on Computer and communications security. ACM, **(2006)**, pp. 79–88.
[12] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren and W. Lou, “Fuzzy keyword search over encrypted data in cloud computing”, in INFOCOM, 2010 Proceedings IEEE, IEEE, **(2010)**, pp. 1–5.

[13] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data", in Data Engineering (ICDE), 2012 IEEE 28th International Conference on. IEEE, **(2012)**, pp. 1156–1167.
[14] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on. IEEE, **(2010)**, pp. 253–262.
[15] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3d chaotic cat maps," Chaos, Solitons & Fractals, vol. 21, no. 3, **(2004),** pp. 749–761.
[16] P. Refregier and B. Javidi, "Optical image encryption using input plane and fourier plane random encoding", in SPIE's 1995 International Symposiumon Optical Science, Engineering, and Instrumentation, International Society for Optics and Photonics, **(1995)**, pp. 62–68.
[17] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, "Enabling search over encrypted multimedia databases", in IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, **(2009)**, pp. 725 418–725 418.

Authors



Yi Zhu, he is currently pursuing his MS in computer science and technology at the School of Computer and Software, in Nanjing University of Information Science & Technology, China. His research interests include information security, cloud security.



Xingming Sun, he received his BS in mathematics from Hunan Normal University, China, in 1984, MS in computing science from Dalian University of Science and Technology, China, in 1988, and PhD in computing science from Fudan University, China, in 2001. He is currently a professor in School of Computer and Software, Nanjing University of Information Science & Technology, China. His research interests include network and information security, digital watermarking.

Zhihua Xia, he received his BE in Hunan City University, China, in 2006, PhD in computer science and technology from Hunan University, China, in 2011. He works as a lecturer in School of Computer and Software, Nanjing University of Information Science & Technology. His research interests include cloud security, and digital forensic.

Naixue Xiong, he received his both PhD degrees in Wuhan University, and Japan Advanced Institute of Science and Technology. He is a Professor in School of Computer Science, Colorado Technical University, USA. His research interests include Security and Dependability, Cloud Computing, Network Architecture, and Optimization Theory.

