A Privacy-preserving Image Retrieval Scheme in Edge Computing Environment

Yiran Zhang [1], Huizheng Geng [1*], Yanyan Xu [2*], Li Su[1], and Fei Liu[3]
[1] China Mobile Research Institute, Beijing 100053, China
[e-mail: zhangyiran@chinamobile.com]
[2] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan university, Wuhan 430079, China
[3] China Mobile Tianjin, Tianjin 300000, China
[*]Corresponding author: Huizheng Geng, Yanyan Xu

Received April 26, 2022; revised November 16, 2022; revised December 25, 2022; accepted January 31, 2023; published February 28, 2023

Abstract

Traditional cloud computing faces some challenges such as huge energy consumption, network delay and single point of failure. Edge computing is a typical distributed processing platform which includes multiple edge servers closer to the users, thus is more robust and can provide real-time computing services. Although outsourcing data to edge servers can bring great convenience, it also brings serious security threats. In order to provide image retrieval while ensuring users' data privacy, a privacy preserving image retrieval scheme in edge environment is proposed. Considering the distributed characteristics of edge computing environment and the requirement for lightweight computing, we present a privacy-preserving image retrieval scheme in edge computing environment, which two or more "honest but curious" servers retrieve the image quickly and accurately without divulging the image content. Compared with other traditional schemes, the scheme consumes less computing resources and has higher computing efficiency, which is more suitable for resource-constrained edge computing environment. Experimental results show the algorithm has high security, retrieval accuracy and efficiency.

Keywords: Image retrieval, Privacy-preserving, Secret sharing, Edge computing.

1. Introduction

Nowadays, the number of graphics and images is growing at a very rapid rate. How to retrieve images from the massive image databases becomes a problem. Content-based image retrieval (CBIR) has been proposed which extracts image features to represent image content, compares the distance and get the query results [1]. CBIR has become one of the fundamental trends in the development of image retrieval technology due to its high efficiency and strong retrieval correlation.

More and more image owners outsource their data to the cloud to reduce the consumption of local storage and computing resources. However, cloud computing adopts the centralized architecture, and the concentration of resources means that the average distance between the users and the cloud server is relatively large, its storage, processing and transmission may bring huge energy consumption and network delay problems [2]. With the increase of the physical distance, the cloud server may not be able to support the delay-sensitive tasks, such as image retrieval and other outsourced tasks. Moreover, it is easy to become the target of attackers when all the images are stored on a single cloud server. There may be irreparable security vulnerabilities once it is destroyed, leading to a single point of failure. Therefore, with the explosive growth in the number of images, image retrieval tasks put forward higher requirements for high bandwidth and low delay, and it is an inevitable trend for computing to sink from cloud to edge.

Edge computing extends cloud computing to the edge of the network. Instead of transmission between the device and the cloud, the data is processed on the edge servers near the user which realizes real-time data processing [3]. The application of the edge server can solve some problems of the outsourced tasks in the cloud environment. On the one hand, the multi-server distributed architecture of edge computing environment offloads the computing burden of centralized cloud server, reducing the possibility of single point of failure. On the other hand, compared with cloud servers, edge servers are closer to users which could minimize service latency and bandwidth. In addition, the parallel processing of multiple servers also improves the efficiency of image retrieval. It's gradually becoming the mainstream paradigm of outsourcing images to edge computing environment to acquire image retrieval services.

Outsourced tasks in edge computing environment can solve some problems of outsourced tasks in cloud computing environment, but multiple edge servers in edge environment are considered "honest but curious" that they may analyze the stored data to learn more information, which may lead to the leakage of image content. To protect data privacy, images should be encrypted locally and then outsourced to the edge environment. However, the encryption operation makes it difficult to maintain the similarity between ciphertext features and plaintext features, which hinders the CBIR operation that are normally performed on plaintext images. Thus, it is essential to develop privacy-preserving image retrieval methods over encrypted domain, which is also known as privacy-preserving content-based image retrieval (PCBIR) [4][5].

The existing PCBIR schemes mainly sort into two main types. The first type is that the image owner constructs the encrypted features and outsources both the cipher-image databases and the encrypted features to the server [6-10,12,13]. These schemes achieve better performance in one aspect of security, accuracy or efficiency, which may damage other performance to a certain extent. The second type is that the image owner only outsources the cipher-image databases to the server, and the server extracts features from the encrypted image and performs the retrieval task [11]. These schemes reduce the computational burden of users,

but they increase the huge computational complexity of the server in the query stage which put forward requirements on the performance of the servers, and task deployment on resource-constrained edge servers may result in longer retrieval time. Therefore, the existing PCBIR scheme is hard to be directly applied to resource-constrained edge computing environment.

It's obvious that there are still some new challenges to be addressed in the edge computing environment. Firstly, compared with the centralized cloud computing, edge computing contains multiple edge servers, which can cooperate to complete computing tasks. In the multi-server PCBIR scheme, how to protect the security of data and retrieval process, achieve a balance between security, efficiency and retrieval accuracy is the biggest challenge. Secondly, the computing power of the edge server is limited compared with cloud server, so it is not suitable to use complex encryption algorithms to perform tasks with high computational complexity in the online query stage. Therefore, we need to redesign the security strategy of image retrieval in edge computing, rather than directly using the existing algorithms.

In this paper, we propose a privacy-preserving image retrieval scheme in edge environment. We generate index shares and trapdoor shares with additive secret sharing technology and upload them to several edge servers which can resist statistical attacks with at least k-1 edge servers. To adapt to distribute computing environment of edge computing, we improve the two-party secure multiplication calculation of original Beaver's multiplication method to k-party secure multiplication calculation so that it can work in k-servers edge environment. Combined with the construction of features, the secure multiplication calculation is converted into the secure Euclidean distance calculation and realize the security similarity calculation by not exposing the real Euclidean distance between the index and trapdoor.

The scheme is mainly divided into offline stage and query stage. In the offline phase, to protect the privacy of images, the image owner extracts features, generates index shares by additive secret sharing technology and uploads them to two or more edge servers respectively together with encrypted image database. In the query phase, the user constructs the trapdoor shares using similar method and uploads them to edge servers respectively. After receiving the trapdoor shares, edge servers calculate the secure Euclidean distance with the improved Beaver's multiplication method and returns the most similar ciphertext images to user. The user decrypts to obtain the plaintext image.

The main contributions in this paper are highlighted as follows:

- This paper presents a PCBIR scheme in edge computing environment, which two or more "honest but curious" servers retrieve the image quickly and accurately without divulging the image content.
- In this scheme, the privacy of image content and image similarity is strictly protected through additive secret sharing and the improved beaver's multiplication protocol, which can resist the collusion attacks of $k(k \geq 2)$ edge servers under COA model and the collusion attacks of $k-1$ edge servers under the KPA model.
- Compared with other traditional PCBIR schemes, the scheme consumes less computing resources in online query stage and has higher computing efficiency, which is more suitable for resource-constrained edge computing environment. In addition, the experiment results prove that the method can realize image retrieval without accuracy loss.

The remainder of this paper is organized as follows. In section 2 we introduce the related work. In section 3 we express the preliminaries. The system model, threat model and design goals are introduced in Section 4. The method is proposed in section 5. And we give the security analysis in section 6. We do the performance evaluation of our scheme in section 7 and give the conclusion in section 8.

2. Related works

At present, the research of PCBIR is mainly focused on cloud computing that the encrypted image and secure index are outsourced to a single cloud server. Randomization technology is a simple and fast binary bit sequence encryption algorithm. Lu et al. [6] propose three image feature protection methods based on randomization technology: bit plane randomization, random projection, and random unary encoding, which have high computational efficiency, but are not safe in practical application. Scrambling encryption technology scrambles the content of an image to a certain extent while retaining some statistical information about the image. Zhang et al. [7] encrypt the image by replacing the DCT coefficient and realize image retrieval on the ciphertext. The scheme can efficiently realize the privacy-preserving image retrieval, while the degree of security is not strong. Order preserving encryption (OPSE) technology can keep the order of ciphertext consistent with that of plaintext, and allow direct sorting of encrypted data. Lu et al. [8] design a secure index scheme using order preserving encryption and random hash function that not only protects data privacy but also provides sorting and searching functions, but this scheme may disclose information under known plaintext attack. Asymmetric scalar-product-preserving encryption (ASPE) is an encryption technique that preserves the inner product between data and query data, making it particularly suitable for retrieval tasks. Kai et al. [9] propose an effective privacy-preserving image retrieval scheme combing secure kNN scheme, vector quantization and ASPE technology, which protects database image features and query image features without revealing the exact distance between them. Homomorphic encryption (HE) is a kind of encryption technology that can perform computation directly on encrypted data. Zhang et al. [10] design the scheme which extracts the features, encrypts the features with homomorphic encryption, and measures the similarity between the encrypted features. Hsu et al. [11] propose a secure retrieval scheme that extracts SIFT features in the encryption domain with privacy-preserving. However, the homomorphic encryption schemes [10][11] result in high computational complexity that make them consume too much time. Local sensitive hash (LSH) can perform fast nearest neighbor search in high-dimensional space, which is often used in large-scale image retrieval. Kuzu et al. [12] generate secure index based on local sensitive hash to realize fast similarity search, which can quickly perform large-scale image retrieval, but may affect the accuracy of retrieval. Secure multi-party computing (SMC) refers to the calculation of statistical functions by multiple parties. All parties can obtain correct results using secure multi-party computing, but no more knowledge can be inferred from the public information. M.Shen et al. [13] design a privacy-preserving image retrieval scheme supporting multiple image sources by using secure multi-party computing technology, but the scheme establishes a fully trusted key management center which reduces the practicability of the scheme.

The above schemes outsource images to the cloud server which can provide large storage resources and high computing resources. But the cloud server is deployed far away from users, it may cause network delay and degradation of service quality. Edge servers are deployed on the edge of the network, which is closer to the user, so it can quickly feedback the results of image retrieval and has better real-time performance. However, the resources of edge servers are limited, which make it difficult for complex encryption algorithms suitable for cloud servers to be directly applied in edge servers.

Some researches consider using secure multi-party computing technology to realize PCBIR schemes in edge computing environment. Yan et al. [14] propose a PCBIR scheme in edge computing environment through data exchange between multiple edge servers, while the scheme must be deployed on a specific number of edge servers, which may lead to problems in the scalability and flexibility. Wang et al. [15] propose a secret sharing homomorphism

scheme, which divide face feature vector into n shares and store them in n servers respectively. Homomorphic technology is used to calculate the sum of the calculation results of any $t(t \leq n)$ servers, while the computational cost is extremely huge caused by the utilization of homomorphic encryption in the query phase, resulting in the lack of practicability.

Table 1 exhibits various comparisons among such algorithms and shows a significant improvement by the proposed scheme [16][17][18]. In general, these schemes are usually difficult to strike a balance between security, efficiency and accuracy. In addition, the use of complex encryption algorithms in the edge server will greatly affect the efficiency of image retrieval due to the limited resources of edge servers, resulting in the PCBIR scheme in cloud environment is no longer suitable for edge computing. Therefore, we propose a lightweight PCBIR scheme suitable for edge computing environment, which achieves a certain balance between security, accuracy and efficiency.

Table 1. Comparative analysis among various algorithms with the proposed scheme

Algorithm	Expediency	Impairments	Comparison with the proposed scheme
Lu et al. [6]	It is high computational efficiency.	It is not safe in practical application, and the accuracy is not very high.	The proposed scheme can strike a better balance between security, efficiency and accuracy.
Zhang et al. [10]	It is high security.	It uses homomorphic encryption which is high computational complexity.	The proposed scheme is very lightweight and suitable for the limited resources of edge computing environment.
M.Shen et al. [13]	It can support multiple image sources.	It establishes a fully trusted key management center which reduces the practicability of the scheme.	The proposed scheme overcomes the problem of low retrieval accuracy and has the same accuracy as the plaintext CBIR.
Yan et al. [14]	It is suitable for edge computing environments.	It must be deployed on a specific number of edge servers which has low scalability and flexibility.	The proposed scheme can be deployed on $k(k \geq 2)$ edge servers which is scalability and flexibility.

3. Preliminaries

3.1 Additive Secret Sharing

The additive secret sharing scheme is a secret sharing technique which ensure the sum of secret shares is equal to the secret. The method splits a secret s into k shares, where each participant will get one share. The secret can be reconstructed when all shares are collected [19]. A typical additive secret sharing system generally consists of two steps:

- The secret sharing algorithm $F_S(s, r) = (s_1, s_2, \dots, s_k)$ takes a secret s and some randomness $r = \{r_1, r_2, \dots, r_{k-1}\}$ chosen at random as input, then outputs k shares $s_1, s_2, \dots, s_k$ of this secret:

$$s_i = \begin{cases} r_i & i = 1, \dots, k-1 \\ s - \sum_{i=1}^{k-1} r_i & i = k \end{cases} \quad (1)$$

- The secret reconstruction function $F_R(s_1, s_2, \dots, s_k) = s$ takes the full set of k shares as input, and the secret can be reconstructed by:

$$s = \sum_{i=1}^{k} s_i \quad (2)$$

where s_i denotes the i^{th} share of the secret s.

3.2 Beaver's multiplication

Beaver's multiplication technology is able to perform multiplication of secret shares with the utilize of triple shares, which is useful for both security and practicality [20][21]. Let $x_i (i \in \{0,1\})$ indicates a additive secret sharing of x, and suppose that triple shares $\{a_i, b_i, c_i\}$ for independent random a, b and $c = ab$ is available. Participant P_i has $\{x_i, y_i\}, \{a_i, b_i, c_i\}$ and the multiplication works of x, y as follows:

1) P_0 calculates $e_0 = x_0 - a_0$, $v_0 = y_0 - b_0$, and sends e_0, v_0 to P_1;
   P_1 calculates $e_1 = x_1 - a_1$, $v_1 = y_1 - b_1$, and sends e_1, v_1 to P_0.
2) P_0 and P_1 calculate $e = e_0 + e_1$, $v = v_0 + v_1$.
3) P_0 calculates $z_0 = v \cdot a_0 + e \cdot b_0 + c_0$, and sends z_0 to P_1;
   P_1 calculates $z_1 = e \cdot v + v \cdot a_1 + e \cdot b_1 + c_1$, and sends z_1 to P_0.
4) P_0 and P_1 calculate $z = z_1 + z_0 = xy$.

3.3 Homomorphic encryption

Homomorphic encryption allows operations to be performed on encrypted data without knowing the private key, and the operation results are the same as those performed on ordinary plaintext data [22]. To reduce the burden of user and ensure the safe generation of triple shares, the proposed scheme uses pallier homomorphic encryption [23] to generate triple shares in the offline phase, which is realized through data exchange between servers without the participation of user.

4. System Design

4.1 System Model

Fig. 1 illustrates the system model of the proposed scheme. There are three types of entities in our system, including image owners, users, and edge servers.

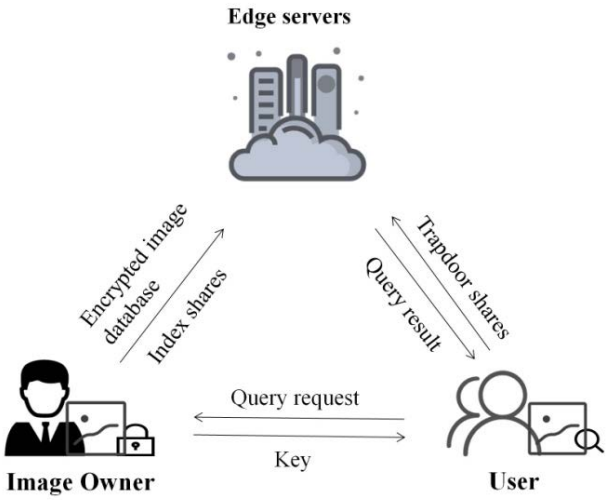

Fig. 1. System model of the scheme

In **Fig. 1**, image owner constructs index shares, encrypts image databases, and then outsources index shares and encrypted image databases to the edge servers respectively. Edge servers store encrypted images databases and index shares, construct triple shares, perform privacy-preserving image retrieval, and return cipher-image. Users construct trapdoor shares, upload trapdoor shares to the edge servers respectively, decrypt cipher-image returned by the edge server and obtain the plaintext query result.

4.2 Threat Model

In general, image owners and users are considered trusted. The security threats of the scheme mainly come from "honest but curious" edge servers and external attackers.

1) External attackers

In the process of transforming data between edge servers and sending data to edge servers by users and image owners, external attackers may eavesdrop information on the communication channel.

2) Edge servers

In this model, the edge servers are considered "honest but curious", which means that the edge servers are following the protocol correctly ("honest"). At the same time, they retain all input from other parties and all intermediate values to infer more knowledge ("curiosity"). Therefore, edge servers will operate normally to avoid being identified by the anomaly detection mechanism, but at the same time they may get the user's privacy information. In this scheme, the privacy of images and image features must be guaranteed. In addition, the original Euclidean distance between images cannot be obtained by the edge server to prevent the edge server from obtaining more information.

4.3 Design goals

1) Image privacy.

We should protect ①image database: the security of image database should be the primary security goal of the scheme which means edge servers cannot obtain image database uploaded by the image owner. ②Index: index should also be encrypted before outsourcing to edge servers. The edge server cannot crack out image features at any time, and the attacker cannot obtain image features by wiretapping channels. For identical image features, the index shares should be completely different. ③trapdoor: The goal of privacy protection is the same as above. ④image similarity: the edge server cannot obtain true Euclidean distance between database image features and query image features. For two identical query features, the Euclidean distance between the query feature and the same databases image feature should be completely different.

2) Retrieval accuracy.

Retrieval accuracy cannot be significantly reduced by adopting privacy protection technology. In this scheme, we ensure that the retrieval accuracy of the proposed scheme is consistent with plaintext image retrieval.

3) Efficiency.

The computing tasks at the user and image owner side should be as few as possible, so as to reduce the computational burden of the user and image owner. Edge servers should undertake most of the computing tasks.

5. The proposed method

5.1 Notations in this section

The secret and its corresponding share used in the proposed scheme are shown in **Table 2**.

Table 2. The secret and its corresponding share used in the proposed scheme

secret	share
triple $a = \sum_{i=1}^{k} a_i$	triple share a_i
triple $b = \sum_{i=1}^{k} b_i$	triple share b_i
triple $c = \sum_{i=1}^{k} c_i = ab$	triple share c_i
encrypted index $f^{'} = \sum_{i=1}^{k} f_i^{'}$	index share $f_i^{'}$
encrypted trapdoor $q^{'T} = \sum_{i=1}^{k} q_i^{'T}$	trapdoor share $q_i^{'T}$
intermediate value $e = \sum_{i=1}^{k} e_i = f' - a$	intermediate share e_i
intermediate value $v = \sum_{i=1}^{k} v_i = q^{'T} - b$	intermediate share v_i
distance $z = \sum_{i=1}^{k} z_i$	distance share z_i

5.2 System overview

Fig. 2 shows the flow chart of the proposed scheme, which consists of three entities: image owner users, and edge servers. For the description, the figure only shows four edge servers, in practical application the number of edge servers can be $k(k \geq 2)$.

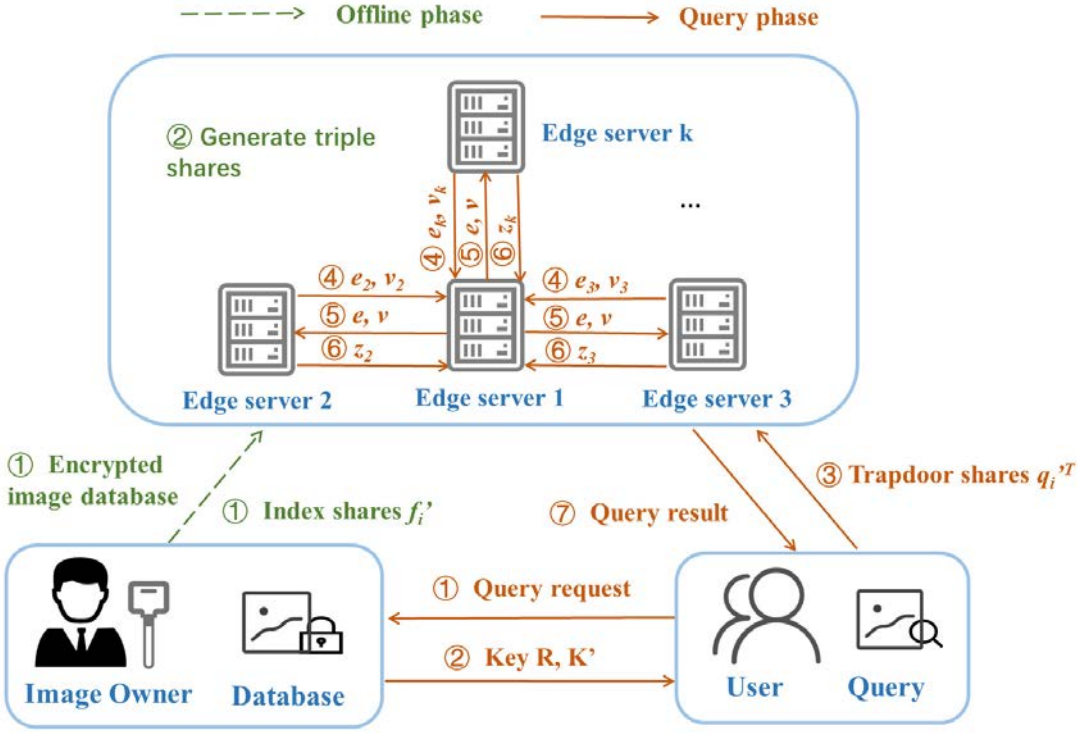

Fig. 2. The flowchart of the scheme

- **Images owners**: They are the provider of the image databases. In the offline phase, the image owner extracts image features, generates index shares, and outsources them to the edge servers together with encrypted images databases.

- **Users**: They are users who want to query images. In the query phase, the user extracts the query image features, generates the trapdoor shares, and sends them to the edge servers. After the edge server returns the cipher-images, they decrypt cipher-images and get plaintext image.
- **The edge servers**: It takes responsibility for performing image retrieval tasks and stores encrypted images, index shares. It contains k edge servers, in which edge server 1 undertakes more computing tasks among all edge servers which needs to aggregate the calculation results and obtains the final similarity ranking.

The specific processing flow is divided into offline phase and query phase as follow:

In the offline phase: ①Image owner constructs index share f_i' $(i = 1,2, \dots, k)$ and encrypts the image database D to obtain the encrypted image database ED. ②The edge servers generate the triple shares and store them which are briefly summarized in the figure and explained in detail below.

In the query phase: ①The user requests authorization from the image owner. ②The image owner confirms the identity of the user, and then returns the index encryption key R and the image decryption key K'. ③After the user is authorized, the user extracts image features, constructs trapdoor shares $q_i'^T$ and uploads them to edge server i respectively. ④After receiving the trapdoor shares uploaded by the user, the edge server i generate intermediate share e_i and v_i, then sends them to edge server 1. ⑤The edge server 1 calculates and broadcasts intermediate value e and v. ⑥The edge server i generates distance shares z_i and sends them to edge server 1. ⑦The edge server 1 sums the distance shares and compares the distance, returns the most similar ciphertext image to the user. The users decrypt to get the plaintext image.

5.3 Offline phase

The main task of this phase is to construct triple shares at the edge servers, build encrypted image database and index shares at the image owner side. The processing flow is as follows:

5.3.1 the edge servers

In the offline phase, the edge server $i(i = 1,2, \dots, k)$ constructs triple shares a_i, b_i, c_i, where a_i and b_i are completely random integers that randomly selected by the edge server i, c_i satisfy $\sum_{i=1}^{k} c_i = \sum_{i=1}^{k} a_i \times \sum_{i=1}^{k} b_i$ which is obtained through the interactive computing between edge servers.

$$\begin{aligned} \sum_{i=1}^{k} c_i &= \sum_{i=1}^{k} a_i \times \sum_{i=1}^{k} b_i \\ &= (a_1 + \cdots + a_k) \times \sum_{i=1}^{k} b_i \\ &= a_1 \times \sum_{i=1}^{k} b_i + \cdots + a_k \times \sum_{i=1}^{k} b_i \\ &= a_1 b_1 + a_1 \sum_{i=2}^{k} b_i + a_2 b_2 + a_2 \sum_{i=1, i\neq 2}^{k} b_i + \cdots \end{aligned} \tag{3}$$

In the (3), $a_i b_i$ can be calculated on edge server i, so the difficulty of constructing triple shares is how to calculate $a_i \sum_{j=1, j\neq i}^{k} b_j$. Here we employ Paillier homomorphic encryption technology to generate $a_i \sum_{j=1, j\neq i}^{k} b_j$. The $Triple()$ algorithm is shown in Algorithm 1.

Algorithm 1 $(c_i) \leftarrow Triple(a_i, b_i)$

For $i = 1:k$
  edge server i generates public key p_i and private key s_i.
  For $j = 1:k$
    If $j \neq i$
      edge server i sends $x_i = Enc_{p_i}(a_i)$ and public key p_i to edge server j.
      edge server j randomly selects $\eta_{i,j}$ and sends $x_{i,j} = x_i^{b_j} Enc_{p_i}(-\eta_{i,j})$ to edge server i.
      edge server i uses private key s_i for decryption and gets $y_{i,j} = a_i b_j - \eta_{i,j}$.
    End If
  End For
  edge server i calculates $c_i = a_i b_i + \sum_{j=1,j\neq i}^{k} y_{i,j} + \sum_{j=1,j\neq i}^{k} \eta_{j,i}$.
End for
Return c_i

The workflow of constructing triple shares a_i, b_i, c_i as follow: First, edge server $i(i = 1,2,\dots,k)$ randomly selects a_i, b_i, generates public key p_i and private key s_i, sends $x_i = Enc_{p_i}(a_i)$ to edge server $j(j = 1,\dots.k, j \neq i)$. Second, after receiving x_i, edge server j randomly selects $\eta_{i,j}$ and sends $x_{i,j} = x_i^{b_j} Enc_{p_i}(-\eta_{i,j})$ to edge server i. Third, edge server i decrypts to obtain $y_{i,j} = a_i b_j - \eta_{i,j}$ using the private key s_i. Finally, edge server i gets $c_i = a_i b_i + \sum_{j=1,j\neq i}^{k} y_{i,j} + \sum_{j=1,j\neq i}^{k} \eta_{j,i}$. In the query phase, the a_i, b_i, c_i will ensure that all processes of data exchange do not reveal privacy to the "honest but curious" servers.

5.3.2 the image owner

In the offline phase, image owner is mainly responsible for generating encrypted image database and index shares that will be outsourced to the edge servers. Encrypted image database are constructed by two steps: executing $\boldsymbol{KeyGen}()$ to generate the image encryption/decryption key, running $\boldsymbol{Enc}()$ to generate encrypted image database. Index shares are built by two steps: executing $\boldsymbol{IndexGen}()$ to generate the index, carrying out $\boldsymbol{IndexShareBuild}()$ to build index shares. The specific steps of each algorithm are explained below:

1）INDEX SHARE BUILD

$\hat{f} \leftarrow \boldsymbol{IndexGen}(I)$. The image owner extracts the color feature [24] of the database image I which can be called database image feature and expressed as $f = (f_1, \dots, f_n)$, where n is the dimension of the feature. Then image owner generates index $\hat{f} = (f_1, \dots, f_n, \|f\|_2^2)^T$, where $\|f\|_2^2 = \sum_{i=1}^{n} {f_i}^2$. $\hat{f} = (f_1, \dots, f_n)^T$

$f_i' \leftarrow \boldsymbol{IndexShareBuild}(\hat{f})$. First, the image owner picks an $(n+1) \times (n+1)$ invertible matrix R at random and generates encrypted index f':

$$f' = \boldsymbol{R}^T \cdot \hat{f} \tag{4}$$

where the matrix R needs to be sent to the user in the query phase.

Second, the image owner takes encrypted index f' and $r_d = \{r_{d,1}, r_{d,2}, \dots, r_{d,k-1}\}$ chosen uniformly at random as input, and outputs k index share f_i' of this secret:

$$F_S(f', r_d) = (f_1', f_2', \dots, f_k') \tag{5}$$

Finally, the image owner sends index share f_i' to edge server i respectively.

2）ENCRYPTED DATABASE GENERATION

$K' \leftarrow \boldsymbol{KeyGen}(1^{\alpha})$. The image owner generates the encryption/decryption key K' of the AES algorithm [25] for encrypting the image database, where α is the parameter for generating the key.

$ED \leftarrow \boldsymbol{Enc}(D, K')$. The image owner encrypts the image database D with image encryption key K' and obtains encrypted image database ED that will be uploaded to the edge server.

5.4 query phase

The user generates the trapdoor shares and sends them to the edge servers respectively. The edge servers execute privacy-preserving image retrieval, which measure the similarity using trapdoor shares and index shares, and return the query result. The user decrypts to get the plaintext image. The processing flow is as follows:

5.4.1 the user

In the query phase, the main task of user is to generate the trapdoor shares and decrypt cipher-image returned by the edge server. Trapdoor shares are built by two steps: executing $\boldsymbol{TrapdoorGen}()$ to generate the trapdoor, carrying out $\boldsymbol{TrapshareBuild}()$ to build trapdoor shares. The user executes $\boldsymbol{Dec}()$ to decrypt the cipher image returned by the edge server and obtain the plaintext query result. The specific steps of each algorithm are explained below:

1）TRAPDOOR SHARE BUILD

$\hat{q} \leftarrow \boldsymbol{TrapdoorGen}(Img_q)$. The user extracts the color feature from the query image Img_q ,which can be called query image feature and denoted as $q = (q_1, \dots, q_n)$, where n represents the dimension of the feature. Then the user constructs the trapdoor $\hat{q} = (-2q_1, \dots, -2q_n, 1)^T$.

$q_i'^T \leftarrow \boldsymbol{TrapshareBuild}(\hat{q})$. First, the user selects a positive number r randomly with uniform probability. The user encrypts $\hat{q}$ to obtain encrypted trapdoor q'^T.

$$q'^T = (r\boldsymbol{R}^{-1} \cdot \hat{q})^T \tag{6}$$

Second, the user takes encrypted trapdoor q'^T and $r_q = \{r_{q,1}, r_{q,2}, \dots, r_{q,k-1}\}$ chosen uniformly at random as input, and outputs k trapdoor share $q_i'^T$ of this secret:

$$F_S(q'^T, r_q) = (q_1'^T, q_2'^T, \dots, q_k'^T) \tag{7}$$

Finally, the user uploads trapdoor share $q_i'^T$ to edge server i respectively.

2）CIPHER IMAGE DECRYPTION

$Img \leftarrow \boldsymbol{Dec}(Img_e, K')$. After receiving the requested cipher image Img_e from the edge server 1, the user gets the query result Img by decrypting Img_e via K'.

5.4.2 the edge servers

After receiving trapdoor shares, the edge servers compare the distance between the index share f_i' and trapdoor share $q_i'^T$ through the interactive calculation between edge servers, and sends the most similar ciphertext images to the user. It mainly consists of two steps: distance share generation and distance reconstruction.

1）DISTANCE SHARE GENERATION

$(z_i) \leftarrow \boldsymbol{DistShareGen}(q_i'^T, f_i', a_i, b_i, c_i)$. After receiving $q_i'^T$ sending from the user, the edge server i executes Algorithm 2 to acquire distance share z_i.

Algorithm 2 $(z_i) \leftarrow DistShareGen(q_i'^T, f_i', a_i, b_i, c_i)$

1：**For** $i = 1{:}\,k$

2： edge server i sends $e_i = f_i' - a_i,\ v_i = q_i'^T - b_i$ to edge server 1；

3：**End for**

4: edge server 1 broadcasts $e = \sum_{i=1}^{k} e_i, v = \sum_{i=1}^{k} v_i$ to the other edge servers；

5：**For** $i = 1{:}\,k$

edge server i sends $z_i = \begin{cases} v \times a_i + e \times b_i + c_i + e \times v & i = 1 \\ v \times a_i + e \times b_i + c_i & i = 2,3,\dots k \end{cases}$ to edge server 1;

6：**End for**

7：**Return** z_i

Edge server i constructs intermediate share e_i, v_i and sends them to edge server 1.

$$e_i = f_i' - a_i \tag{8}$$

$$v_i = q_i'^T - b_i \tag{9}$$

After receiving e_i, v_i from edge server i, edge server 1 constructs intermediate value $e = \sum_{i=1}^{k} e_i, v = \sum_{i=1}^{k} v_i$ and broadcasts them to the other edge servers.

After receiving e, v from edge server 1, edge server i generates distance share z_i and sends them to edge server 1.

$$z_i = \begin{cases} v \times a_i + e \times b_i + c_i + e \times v & i = 1 \\ v \times a_i + e \times b_i + c_i & i = 2,3,\dots k \end{cases} \tag{10}$$

2）DISTANCE RECONSTRUCTION

$(Img_e) \leftarrow \boldsymbol{DistRecon}(z_i)$. After receiving distance share z_i from the other edge servers, edge server 1 reconstructs distance z. The smaller z is, the more similar the database image is to the query image. According to this principle, the retrieved ciphertext images Img_e will be found and sent to the user.

$$z = \sum_{i=1}^{k} z_i = q'^T f' \tag{11}$$

Proof Depending on the nature of additive secret sharing, we know $a = \sum_{i=1}^{k} a_i, b = \sum_{i=1}^{k} b_i, c = \sum_{i=1}^{k} c_i, e = \sum_{i=1}^{k} e_i = f' - a, v = \sum_{i=1}^{k} v_i = q'^T - b$, then

$$\begin{aligned} z &= \sum_{i=1}^{k} z_i \\ &= ev + v\sum_{i=1}^{k} a_i + e\sum_{i=1}^{k} b_i + \sum_{i=1}^{k} c_i \\ &= ev + va + eb + c \\ &= (q'^T - b)(f' - a + a) + (f' - a)b + c \\ &= q'^T f' - f'b + f'b - ab + ab \\ &= q'^T f' \end{aligned} \tag{12}$$

We have $q'^T f' = (r\boldsymbol{R}^{-1} \cdot \hat{q})^T (\boldsymbol{R}^T \cdot \hat{f}) = r(\|f\|_2^2 - 2\sum_{i=1}^{n} f_i q_i) = r(\|q - f\|_2^2 - \|q\|_2^2)$, where $\|q - f\|_2^2 = \sum_{i=1}^{n} (q_i - f_i)^2$, $\|q\|_2^2 = \sum_{i=1}^{n} {q_i}^2$. For each query, the constant is $\|q\|_2^2$ and r, variable is $\|q - f\|_2^2$. It is obvious that the ranking of z maintains the ranking of original Euclidean distance $\|q - f\|_2^2$ between query image features q and database image features f. Moreover, for two identical query inputs, distance z between the images is completely different, so as to prevent the edge servers from obtaining the similarity information between the images. The improved Beaver's multiplication method transforms the multiplication calculation into the distance calculation, ensures that the original Euclidean distance sorting is completely retained and realizes the security similarity calculation without exposing the real Euclidean distance.

6. Security Analysis

The security of the proposed scheme is analyzed under ciphertext-only attack(COA) [26], known-plaintext attack(KPA). The private information we need to protect includes image database, index, trapdoor, image similarity. The AES technology is used to protect image databases, which has been proved to be a secure encryption algorithm [27]. In a system with k edge servers, the security of index, trapdoor and image similarity is discussed as follow.

1) Analysis under COA model

In COA model, the attackers can access the ciphertext stored on the edge servers, but do not know the plaintext and keys.

Theorem 1 Under the premise that all k edge servers or fewer edge servers are corrupted by a semi-honest attacker, the proposed scheme can ensure that the edge servers cannot obtain the private information of the user and the image owner under COA model.

Proof Assuming that k edge servers are corrupted by the attacker, we analyze their security from the following three aspects:

- Suppose k index shares are corrupted, the attacker $\mathcal{A}$ can obtain the encrypted index f' by aggregating index shares. However, encrypted index refers to the index is encrypted using the invertible matrix R. Therefore, the only way for an attacker $\mathcal{A}$ to get the value of the index is to enumerate all possible values. We assume that the dimension of the index is $n+1$, and the range of the image features is T. The computational complexity of an attacker $\mathcal{A}$ directly simulating index is T^{n+1}, so the security strength is $log_2(T^{n+1})$. The security of trapdoor is analyzed in a similar way.
- Suppose k intermediate shares are corrupted, the attacker $\mathcal{A}$ can obtain intermediate values e, v by aggregating intermediate shares and triples a, b, c by aggregating triple shares. So the attacker $\mathcal{A}$ can get the encrypted index $f' = e + a$, the encrypted trapdoor $q'^T = v + b$. But encrypted indexes do not reflect real information of index, the security strength is $log_2(T^{n+1})$.
- Suppose k distance shares are corrupted, the attacker $\mathcal{A}$ can obtain $z = r(\|q - f\|_2^2 - \|q\|_2^2)$. Since q and r are unknown, the true Euclidean distance cannot be obtained. The attacker $\mathcal{A}$ tries to get information of true Euclidean distance by exhausting all possibilities, however even if all possible values are exhausted, the correct value cannot be identified from them.

Therefore, the private information of the user and the image owner can be protected.

2) Analysis under KPA model

In KPA model, the attacker $\mathcal{A}$ can access several pairs of plaintext and the corresponding ciphertext.

Theorem 2 Under the premise that $k-1$ edge servers or fewer edge servers are corrupted by a semi-honest attacker, the proposed scheme can ensure that the edge servers cannot obtain the private information of the user and the image owner under the KPA model.

Proof Assuming that $k-1$ or less edge servers are intruded by the attacker $\mathcal{A}$. Meanwhile, the attacker $\mathcal{A}$ extracts the feature from the query image as a query user, $k-1$ secret shares corresponding to this feature are obtained by the attacker $\mathcal{A}$. However, the nature of the additive secret share explains that no more than $k-1$ secret shares can not reveal any true information about the secret. Therefore, the attacker $\mathcal{A}$ cannot obtain any private information.

7. Performance Evaluation

In this chapter, we give the experimental results and evaluate the performance of the proposed scheme in terms of the security evaluation, retrieval accuracy and efficiency. Experiments are performed on two datasets, one dataset is corel1000 [28], another dataset is caltech1000 database. The two image datasets have 10 types of pictures, 100 pictures of each type. The retrieval results of the proposed scheme are compared with several typical privacy-preserving image retrieval schemes, including Lu's three schemes [6], Homomorphic encryption scheme [10]. All experiments are implemented by MATLAB.

7.1 Evaluation of Security

1) The autocorrelation function

The autocorrelation function describes the correlation degree of neighboring signal [29], the autocorrelation function $R(T)$ is shown as follows:

$$R(T) = \sum_{i-1}^{n} x(i)x(i-T) \tag{13}$$

where n is the length of the signal, T is the delay of the signal.

The autocorrelation function for the plaintext feature, the encrypted features of three methods proposed by Lu [6], the index share which is the encryption feature of the proposed scheme, and the encrypted features in homomorphism scheme [10] are shown in **Fig. 3**.

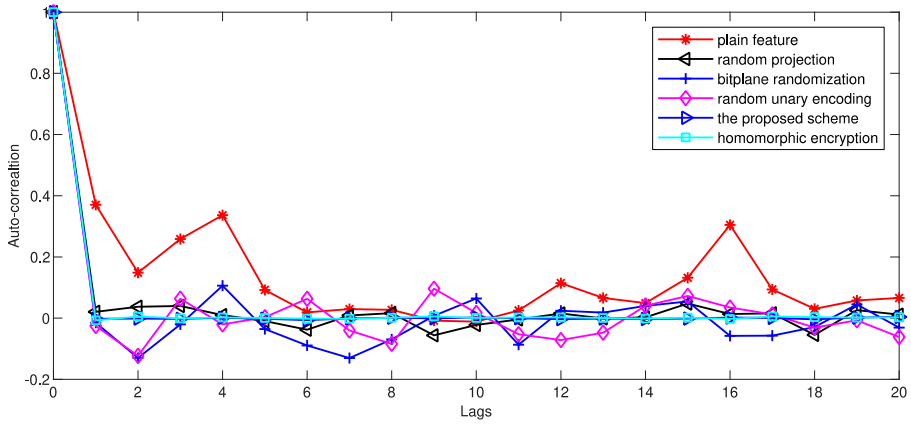

Fig. 3. Autocorrelation function of encrypted features

From results, we can see that the plaintext features have strong correlation, which indicates that there is a strong correlation between adjacent features. It is clear that the autocorrelation of the index share is obviously lower than that of encrypted features built by Lu's three methods. The autocorrelation of the index share is almost zero in the proposed method, which means that attackers cannot infer any plaintext feature information from the shares. It is generally considered that homomorphic encryption algorithm has strong security, and its autocorrelation function fluctuates near zero. The results show that the autocorrelation of index share is similar as that of encrypted features generated by homomorphic encryption, which indicates the share has high security. In addition, the autocorrelation ability of the proposed scheme will be further weakened and the security will be further enhanced with the expansion of the range of random numbers and the increase of the number of edge servers.

2) Information entropy

Information entropy represents the uncertainty of encrypted features, which is defined as follows:

$$H = -\sum_{i=0}^{n} p(x(i)) \times log_2 p(x(i)) \tag{14}$$

where $x(i)$ is the encrypted feature, n is the length of $x(i)$, $p(x(i))$ is the probability of $x(i)$. The entropy for the index share of the proposed scheme, the encrypted features of three methods proposed by Lu [6], the encrypted features in homomorphism scheme [10], plaintext features and random data are shown in **Table 3**.

Table 3. Information entropy of encrypted features

Method	Information entropy
Proposed scheme	6.6433
Random projection	5.6988
Bit-plane randomization	5.6545
Randomized unary encoding	5.6387
Homomorphic encryption	6.6434
Plaintext features	1.3410
Random data	6.6475

The experimental results show that the entropy of plaintext features is low, thus there is a strong internal correlation between plaintext features. The entropy of the share in this paper is almost the same as that of encrypted features built by homomorphic encryption, and is significantly higher than that of the encryption features obtained by other schemes, which shows that the distribution of share is very random and has high security. It is considered that homomorphic encryption has high computational complexity and low efficiency, and the time consumed of the proposed scheme is significantly less than homomorphic encryption which proves the scheme is in fact practical.

7.2 Evaluation of Retrieval Accuracy

The most common evaluation measures of retrieval accuracy in image retrieval are $precision$ and $recall$ usually presented as a $precision\ vs\ recall\ graph$ (PR graph) [30].

$$precision = \frac{No.relevant\ images\ retrieved}{Total\ No.images\ retrieved}$$

$$recall = \frac{No.relevant\ images\ retrieved}{Total\ \ No.relevant\ images\ in\ the\ collection}$$

In the experiments, a group of 100 query images containing 10 categories of images are retrieved in the Corel1000 dataset and Caltech1000 dataset. We compare the proposed method with the Lu's three method [6] and homomorphic encryption scheme [10], and the comparison results of PR graph are shown in **Figs. 4, 5**.

It is obvious that the retrieval result of the proposed scheme performs almost the equal with homomorphic encryption scheme, better than Lu's three methods. Homomorphic encryption scheme is usually considered as a method with approximately no loss of accuracy, which shows that the scheme can achieve accurate image retrieval.

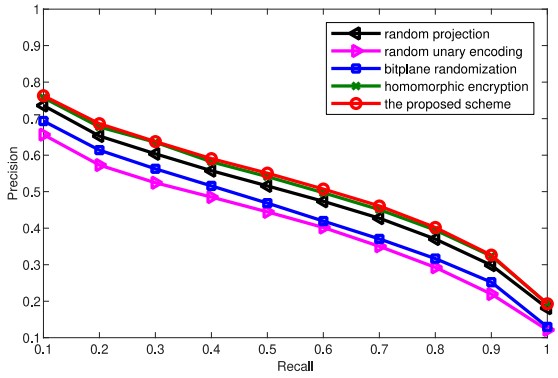

Fig. 4. The accuracy comparison in Corel1000

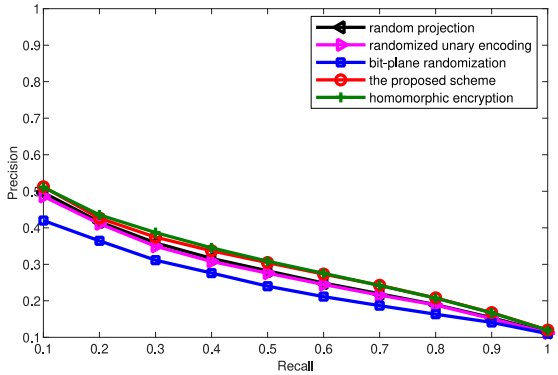

Fig. 5. The accuracy comparison in Caltech1000

We also compare our scheme with plaintext image retrieval, the results are shown in **Figs. 6, 7**. The results demonstrate that the retrieval accuracy of the method is almost consistent with that of plaintext image retrieval. The scheme requires that the number of edge servers should be at least two. On the premise of meeting this condition, the image retrieval accuracy will not decrease with the increase of the number of edge servers. Regardless of the number of edge servers, the precision of the scheme is consistent with that of plaintext image retrieval.

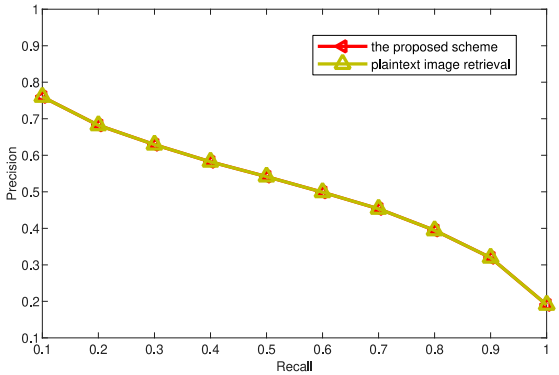

Fig. 6. The accuracy comparison of the scheme and plaintext image retrieval in Corel1000

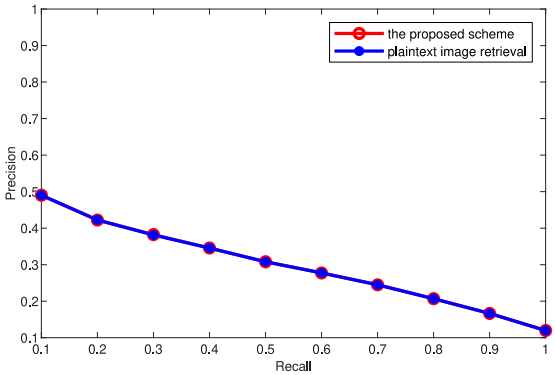

Fig. 7. The accuracy comparison of the scheme and plaintext image retrieval in Caltech1000

7.3 Evaluation of efficiency

We analyze the efficiency of the scheme from the perspective of computational complexity and interaction time.

7.3.1 Computational complexity

Assume n denotes the size of the data, k represents the number of the edge server, g is the number of bit planes to encrypt, m is the dimension of projected features, M is the maximum possible value in all the feature vectors, then the computational complexity results are shown in **Table 4**.

Table 4. Time complexity comparison of different methods

Method	Time complexity
Proposed scheme	$O(kn)$
Random projection	$O(mn)$
Bit-plane randomization	$O(gn)$
Randomized unary encoding	$O(mnM)$
Homomorphic encryption	$O(n^4(n^n)^2)$

The experimental results express that the time complexity of the homomorphic encryption method is very high. The computational complexity of the proposed scheme is lower than Homomorphic encryption, Randomized unary encoding and Bit-plane randomization. The computational complexity of random projection is $O(mn)$ which is relatively low. Compare with random projection, the time complexity comparison of the proposed scheme is interrelated to the number of the edge server k. The larger the value of k, the greater the computational complexity. However, the accuracy of random projection is relatively low, and its accuracy decreases significantly with the increase of the size of image databases.

7.3.2 Evaluation of interaction time

In addition to computing time, we discuss the overhead of interacting with multiple edge servers. Since all the work in the offline phase can be completed before the user initiates the query, this section mainly discusses the interaction cost in the query phase, and the interaction time includes three parts:

1) The user sends the trapdoor shares to k edge servers respectively. This process can happen almost simultaneously.

2) The interactive computing between edge server 1 and the other edge servers for generating distance shares. This process can be regarded as almost simultaneous transmission. Moreover, the communication time of this process does not increase significantly with the increase of the edge server.
3) The edge server 1 returns the query results to the user. This part of the interaction time must be consumed.

Although there is a certain amount of interaction time, most communications can be executed almost at the same time. With the development of 5G communication technology, it can transmit data quickly to ensure the rapid execution of image retrieval.

We conduct experiments on Corel1000 datasets for different encryption methods in a query task, obtain the time comparison results in the offline phase and the query phase shown in **Table 5**. In the experiment of the proposed scheme, the number of the edge server used is 3, and the number of triple shares generated is one set.

Table 5. Comparison results of time consumption

Method	Time in offline phase	Time in query phase
Proposed scheme	424.546135	0.715641
Random projection	414.678223	0.667861
Bit-plane randomization	571.736372	1.839401
Randomized unary encoding	818.392223	1.041243
Homomorphic encryption	916.567323	28.247373

The experimental results demonstrate that the proposed scheme takes less time than homomorphic encryption, randomized unary encoding, bit-plane randomization in the offline phase and query phase. The scheme is slightly slower than the random projection which is a technique to randomize characteristic disturbances. The performance of random projection in security and retrieval accuracy is obviously worse than that of our scheme, especially its security has been proved to be relatively weak, thus our scheme achieves the balance among security, accuracy and retrieval efficiency to some extent in edge computing environment.

8. Conclusion

In order to solve the problems in the existing PCBIR schemes on cloud computing, such as huge energy consumption, network delay and single point of failure, we present a privacy-preserving image retrieval scheme in edge computing environment, suitable for the distributed and resource-constrained characteristics of edge servers. We encrypt image features with additive secret sharing, so the attacker must corrupt at least $k-1$ edge servers to crack the system, which greatly enhances the security of the system. We also improve the Beaver's multiplication to evaluate the similarity of images and avoid exposing original Euclidean distance to the edge servers. Security analysis and experimental results demonstrate the proposed method achieves the balance between security, accuracy and efficiency. Future work will develop the efficiency of preprocessing in the offline stage, and better adapt to the edge computing environment.

References

[1] Smeulders, M. Worring, S. Santini, "Content-based image retrieval at the end of the early years," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, pp. 1349-1380, December, 2000. Article (CrossRef Link)

[2] X. Zhang, Q. Chen, X. Peng, "Differential privacy-based indoor localization privacy protection in edge computing," in *Proc. of 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, 2019. Article (CrossRef Link)

[3] Z. Yan, J. Xue, C.W. Chen, "Prius: hybrid edge cloud and client adaptation for HTTP adaptive streaming in cellular networks," *IEEE Trans. Circuits Syst. Video Technol*, Vol. 27, pp. 209–222, 2017. Article (CrossRef Link)

[4] J. Shashank, P. Kowshik, K. Srinathan, "Private content based image retrieval," in *Proc. of 2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008. Article (CrossRef Link)

[5] Y. Xu, J. Gong, L. Xiong, "A privacy-preserving content-based image retrieval method in cloud environment," *Journal of Visual Communication & Image Representation*, vol. 43, pp. 164-172, 2017. Article (CrossRef Link)

[6] W. Lu, Varna, A. Swaminathan, "Secure image retrieval through feature protection," in *Proc. of 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009. Article (CrossRef Link)

[7] X. Zhang, H. Cheng, "Histogram-based retrieval for encrypted JPEG images," in *Proc. of 2014 IEEE China Summit & International Conference on Signal and Information Processing*, pp. 446-449, 2014. Article (CrossRef Link)

[8] W. Lu, A. Swaminathan, Varna, "Enabling search over encrypted multimedia databases," *Media Forensics and Security*, pp, 404-414, 2009. Article (CrossRef Link)

[9] H. Kai, X. Ming, S. Fu, "Efficient Privacy-Preserving Content-Based Image Retrieval in the Cloud," in *Proc. of International Conference on Web-age Information Management*, pp. 28-39, 2016. Article (CrossRef Link)

[10] Y. Zhang, L. Zhuo, Y. Peng, "A secure image retrieval method based on homomorphic encryption for cloud computing," in *Proc. of 2014 19th International Conference on Digital Signal Processing*, pp. 269-274, 2014. Article (CrossRef Link)

[11] C. Y. Hsu, C. S. Lu, S. C. Pei, "Image feature extraction in encrypted domain with privacy-preserving SIFT," *IEEE transactions on image processing*, vol. 21, no. 11, pp. 4593-4607, 2012. Article (CrossRef Link)

[12] M. Kuzu, M. S. Islam, M. Kantarcioglu, "Efficient similarity search over encrypted data," in *Proc. of 2012 IEEE 28th International Conference on Data Engineering*, 2012. Article (CrossRef Link)

[13] M. Shen, G. Cheng, L. Zhu, "Content-based multi-source encrypted image retrieval in clouds with privacy preservation," *Future Generation Computer Systems*, vol.109, pp. 621-632, 2020. Article (CrossRef Link)

[14] Y. Yan, Y. Xu, Y. Zhang, "Privacy-preserving content-based image retrieval in edge environment," *Cluster Computing*, vol. 25, pp. 363-381, 2021. Article (CrossRef Link)

[15] X. Wang, H. Xue, X. Liu, "A privacy-preserving edge computation-based face verification system for user authentication," *IEEE Access*, vol. 7, pp. 14186-14197, 2019. Article (CrossRef Link)

[16] S. Ashraf, "Culminate Coverage for Sensor Network Through Bodacious-instance Mechanism," i-*manager's Journal on Wireless Communication Networks*, vol. 8, no. 3, pp. 1-9, 2019. Article (CrossRef Link)

[17] S.Ashraf, O.Alfandi, A.Ahmad, A. M.Khattak, B.Hayat, K. H. Kim, "Bodacious-Instance Coverage Mechanism for Wireless Sensor Network," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–11, 2020. Article (CrossRef Link)

[18] S.Ashraf, T.Ahmed, S.Saleem, "NRSM: node redeployment shrewd mechanism for wireless sensor network," *Iran Journal of Computer Science*, vol. 4, pp. 171–183, 2021. Article (CrossRef Link)

[19] Q. Li, I. Cascudo, M. G. Christensen, "Privacy-preserving distributed average consensus based on additive secret sharing," in *Proc. of 2019 27th European Signal Processing Conference*, 2019. Article (CrossRef Link)
[20] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *Proc. of Annual International Cryptology Conference*, vol.576, 1992. Article (CrossRef Link)
[21] M. Keller, "MP-SPDZ: A versatile framework for multi-party computation," in *Proc. of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1575-1590, 2020. Article (CrossRef Link)
[22] T. Sridokmai, S. Prakancharoen, "The homomorphic other property of Paillier cryptosystem," in *Proc. of 2015 International Conference on Science and Technology (TICST)*, pp. 356-359, 2015. Article (CrossRef Link)
[23] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," *EUROCRYPT 1999: Advances in Cryptology — EUROCRYPT '99*, vol. 1592, pp. 223-238, 1999.
[24] J. P. Eakins, M. E. Graham, "Content-based image retrieval, a report to the JISC Technology Applications programm," 1999.
[25] J. Daemen, V. Rijmen, The Design of Rijndael The Advanced Encryption Standard (AES): The Advanced Encryption Standard (AES).
[26] H. V. Tilborg, S. Jajodia, *Encyclopedia of Cryptography and Security*, Springer US, 2011.
[27] Fan L, Luo J, Liu H, et al, "Data security concurrent with homogeneous by AES algorithm in SSD controller," *IEICE Electronics Express*, vol. 11, 2014. Article (CrossRef Link)
[28] J. Li, J. Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 9, pp. 1075-1088, 2003. Article (CrossRef Link)
[29] Haykin, Simon, *Nonlinear methods of spectral analysis*, Springer Science & Business Media, 2006.
[30] H. Müller, W. Müller, D. M. G. Squire, "Performance evaluation in content-based image retrieval: overview and proposals," *Pattern recognition letters*, vol. 22, no. 5, pp. 593-601, 2001. Article (CrossRef Link)

Yiran Zhang, is a researcher at China Mobile Research Institute, Beijing, China. She received the B.E. degree in computer science and technology from Zhengzhou University, China, in 2018, the M.S. degree in computer application technology from Wuhan University, China, in 2021. Her current research is data security, multimedia information security. Email: zhangyiran@chinamobile.com.

Huizheng Geng, is a technical manager at the China Mobile Research Institute. He received his M.S. in information security from Shandong University in 2014. He has 8 years' experience in the field of data security. He is now a technical manager at the China Mobile Research Institute. His current research is data security and network security. Email: genghuizheng@chinamobile.com.

Yanyan Xu, is a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University. She received the B.E. degree from the Xi'nan Institute of Technology, China, in 1995, the M.E. degree in electrical engineering from the Hubei University of Technology, China, in 2000, and the Ph.D. degree in communication and information system from Wuhan University, China, in 2007. Her current research is multimedia information security and multimedia communication systems. Email: xuyy@whu.edu.cn.

Li Su, is the Vice Director of the Security Technology Department at the China Mobile Research Institute. He received his Ph.D. in information security from Huazhong University of Science and Technology in 2008. He has 14 years' experience in the field of information security of telecommunications and holds the certificate of Certified Information Systems Security Professional (CISSP). His current research is network security and information security. Email: suli@chinamobile.com.

Fei Liu, is the head of the Network Security Laboratory at the China Mobile Tianjin. He received his M.S. in software engineering from Beijing University of Posts and Telecommunications in 2012. He has 10 years' experience in the field of information security of telecommunications and holds the certificate of Certified Information Security Professional (CISP). His current research is network security and information security. Email: liufei@tj.chinamobile.com.