An Accurate and Privacy-preserving Retrieval Scheme over Outsourced Medical Images

Dan Zhu, Hui Zhu, *Senior Member, IEEE,* Xiangyu Wang, Rongxing Lu, *Fellow, IEEE,* and Dengguo Feng

Abstract—With the rapid advancement in medical imaging techniques, Content-Based (medical) Image Retrieval (CBIR), which can assist in disease diagnosis, has gained much attention in both academia and industry. However, due to patients' sensitive information involved in medical images, privacy-preserving CBIR is a challenge worth exploiting. Though several privacy-preserving CBIR schemes have been put forth, they can only resist known-background attack (KBA), and do not suffice for protecting the image privacy in outsourced settings. In this paper, aiming at the above challenge, we first design a novel Privacy-preserving Mahalanobis Distance Comparison (PMDC) method to improve the accuracy of medical images retrieval. Then, combined with the Mahalanobis distance based Fuzzy C-Means (FCM-M) algorithm, a scheme named TAMMIE is proposed to achieve accurate and privacy-preserving medical image retrieval over encrypted data. With TAMMIE, an image owner can securely outsource the images and indexes to a cloud server, and query users can request retrieval services from the cloud server while keeping their queries private. Detailed security analysis shows that our proposed schemes are secure under the attack stronger than KBA. Furthermore, thorough empirical experiments conducted on two real-world and one synthetic datasets also demonstrate the efficiency of TAMMIE.

Index Terms—Medical images, privacy-preserving, content-based image retrieval, Mahalanobis distance, Fuzzy C-Means.

1 Introduction

The extensive applications of medical imaging techniques have recently triggered the explosive growth of medical image data. To leverage these data and mine the potential value, similar images retrieval techniques perform more critical than ever. Therefore, Content-Based Image Retrieval (CBIR) technique, which can automatically extract image visual features (e.g., colours, textures, and shapes) and find similar images, has attracted widespread researchers' attention in the medical field [1]. Applying CBIR to retrieve similar medical images can help physicians find out the previous similar cases [2], and make an assessment for the new patient's health status.

Meanwhile, driven by the heavy local computation and communication burden, the medical institution tends to outsource medical images and indexes to a cloud server for cost-saving and services providing. However, the cloud server cannot be fully trusted and medical data involves the patients' private information. As a result, the flourish of similar medical image retrieval service is still confronted with some serious obstacles, including the privacy and security issues of medical images and indexes. For one thing, the medical images usually hide the patients' histories of diseases. For another, the extracted indexes can reveal the characteristics of medical images. Once directly outsourcing them to a cloud server with ulterior motives, sensitive patients' personal data is likely to be leaked under the temptation of commercial interests. In a word, the introduction of the cloud server will increase the privacy risk of medical image data although it reduces the medical institution's computation tasks.

To address the above-mentioned challenges, a number of privacy-preserving CBIR (PPCBIR) schemes have been proposed. In 2009, Wong *et al.* [3] proposed a secure K-Nearest Neighbors (KNN) algorithm, which can compare Euclidean distance over encrypted vectors efficiently. Motivated by their work, many efficient PPCBIR schemes [4]–[9] (denoted as KNN-PPCBIRs) have been put forth. Nevertheless, according to the formal security analysis in [10], secure KNN is proven to be actually insecure against even Ciphertext-Only Attack (COA[1]). Thus KNN-PPCBIRs have no ability in resisting the security threats in an outsourcing environment. After that, Yuan *et al.* [11] constructed a secure K-means framework, in which a Privacy-preserving Euclidean Distance Comparison (PEDC) technique was proposed. Combined PEDC with a key conversion protocol, Wang *et al.* [12] and Li *et al.* [13] presented two PPCBIR schemes to support multi-key multi-user settings. Through their security analysis, both of these two schemes can resist Known-Background Attack (KBA[2]), which is stronger than COA. *In conclusion, most of the existing schemes sacrifice data privacy to pursue higher efficiency.* Moreover, the clustering techniques (e.g., K-means [14], CAK-means [15]) used in the existing PPCBIR schemes [6], [7], [16] to *speed up the retrieval usually reduce search accuracy.* Besides, the similarity metrics selected by these work [4]–[9], [12], [13], [16] are Euclidean distance or Hamming distance, which are suitable for common image datasets, but they may not be the best

- *D. Zhu, H. Zhu, X. Wang are with the National Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, 710071, China (e-mail: zhudan@stu.xidian.edu.cn, zhuhui@xidian.edu.cn, xywang_xidian@163.com, corresponding author: Hui Zhu).*
- *R. Lu is with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada (e-mail: rlu1@unb.ca).*
- *D. Feng is with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, 100190, China (e-mail: fengdg@263.net).*

1. The attacker can only access the encrypted data.
2. Apart from the ciphertexts, the attacker can use statistical information to deduce specific contents in a query and learns other background information, but it cannot obtain plaintext-ciphertext pair.

choice for medical images due to *they do not take dimensional correlation into consideration*.

Contribution. In this paper, based on an existing matrix encryption technique [17], we first design a privacy-preserving Mahalanobis distance (MD) comparison method called PMDC. Then, by introducing the MD based Fuzzy C-Means (FCM-M) algorithm and improving PMDC, we propose an accurate and privacy-preserving similar medical image retrieval scheme, named TAMMIE. Specifically, the main contributions of this paper are listed as follows.

- *Novel medical image retrieval scheme.* We first propose PMDC, which supports Mahalanobis distance comparison over ciphertexts, as the basic method of medical image retrieval. Then, we present TAMMIE to speed up the retrieval and improve the search accuracy. Specifically, TAMMIE introduces FCM-M algorithm to cluster similar images so that the search efficiency and accuracy can be improved. To suit the search method of FCM-M, we improve PMDC to achieve distance comparison of multiple clusters with different covariance matrices.
- *Privacy preserving.* TAMMIE keeps all medical images and indexes (queries) confidential during the storage and retrieval process through symmetric encryption and matrix encryption techniques. Security analysis shows that our work can achieve indistinguishable secure under the known-plaintext attack model, which means TAMMIE possesses stronger security level than most of existing similar schemes.
- *High accuracy.* The combination of MD metric and FCM-M algorithm makes our proposed retrieval scheme achieve higher accuracy. Experimental results using two real-world medical image datasets show that the accuracy of TAMMIE performs better than that of two state-of-the-art similar schemes, in which Euclidean and Hamming distances are respectively employed as the similarity metrics.
- *Availability.* We implement TAMMIE and two comparison schemes with Python programming language, and conduct extensive performance evaluations on two real-world medical images datasets together with one randomly generated synthetic dataset. The evaluation results demonstrate the efficiency of our TAMMIE is comparable with existing schemes while providing better security.

The remainder of this paper is organized as follows. At first, we review the related work in Section 2 and clarify the models and privacy requirements in Section 3. Then, some basic knowledge used in our work are introduced in Section 4. After that, we give the details of PMDC in Section 5 and construct the framework of TAMMIE in Section 6. Next, the security and performance of the proposed schemes are analyzed in Section 7 and Section 8, respectively. Finally, we draw a conclusion in Section 9.

2 Related work

In this section, we briefly review some related work on Privacy-Preserving Content-Based (medical) Image Retrieval (PPCBIR), which enables the data owners to store the encrypted images and indexes in the cloud server, and supports similar image retrieval over ciphertext domain.

To the best of our knowledge, the first endeavor on PPCBIR was made in [18], based on secure inverted index and secure min-Hash algorithm, Lu *et al.* proposed two secure indexing schemes, both of them can achieve the similarity of two images by computing the Jaccard index between their encrypted feature vectors. Meanwhile, Lu *et al.* [19] also combined signal processing (e.g., bit-plane randomization, random projection, and randomized unary encoding) and cryptographic techniques (e.g., XOR and random permutation) to investigate three feature protection schemes, which can be used as building blocks to achieve PPCBIR over large encrypted image databases. In view of the fact that scale-invariant feature transform (SIFT) [20] can be used for image feature extraction, Hsu *et al.* [21] proposed a privacy-preserving SIFT method based on homomorphic encryption to address the secure problem encountered in the outsourced environment. After that, the improved homomorphic encryption techniques are employed in [22], [23] to encrypt indexes, although they achieved more efficient retrieval than conventional privacy-preserving schemes, the computation overhead is too high to be suitable for practical applications.

Compared with homomorphic encryption based techniques [24], the work in [25] proved that the feature/index randomization-based techniques perform better in term of computational and communication cost. In order to ensure the accuracy and efficiency of image retrieval, Xia *et al.* [4] first designed an efficient PPCBIR scheme in the cloud computing scenario by introducing a secure KNN technique [3] to encrypt indexes, and then constructed pre-filter tables to improve the retrieval efficiency based on Locality-Sensitive Hashing (LSH) technique. With secure KNN, Xia *et al.* [5] presented another efficient and copy-deterrence PPCBIR scheme similar to [4], which utilizes watermark extraction to trace unlawful query user to improve the security. Analogously, Yuan *et al.* [6] extracted Fisher vector as indexes and used K-means clustering to narrow the search range, they achieved a lightweight PPCBIR scheme over encrypted data based on secure KNN, too. In addition, PPCBIR schemes [7]–[9] were also proposed based on secure KNN. [7] employed a deep learning model, Convolutional Neural Networks (CNN), to improve the accuracy of retrieval, and constructed an encrypted hierarchical index tree to speed up the query phase. Li *et al.* [8] proposed a PPCBIR scheme in multi-user settings by using the polynomial-based access strategy and proxy re-encryption technique, and achieved malicious search user tracing through the watermark technique. Further considering the settings of multi-owner, Tong *et al.* [9] presented a verifiable fine-grained encrypted image retrieval scheme, which is capable of supporting efficient fine-grained access control and result verification simultaneously. However, secure KNN has been proven is insecure under the threat model in which the adversary only knows ciphertext [10]. Aiming at constructing a PPCBIR scheme with higher level of security, Wang *et al.* [12] proposed a practical outsourced image retrieval framework based on a secure comparison technique called PEDC designed in [11], and their scheme supported unshared key search. To take full advantage of data structure, Li *et al.* [16] re-designed

the hierarchical index tree based on CAK-means clustering algorithm and efficiently performed the retrieval process over encrypted indexes.

Meanwhile, with the development of emerging blockchain technique and deep learning (DL) models, Shen *et al.* [26] proposed a privacy-preserving blockchain-based medical image retrieval system which can be applied to the multi image data providers scenarios. Devaraj *et al.* [27] presented a secure image archival and retrieval system using DL and multiple share creation schemes, deep learning is responsible for feature extraction and multiple share creation schemes preserves the data privacy. And Vepakomma *et al.* [28] designed a novel differentially private for supervised manifold learning, which can be used in PPCBIR. Besides, [29]–[31] also paid attention to extracting feature vectors with quicker speed and more accuracy with DL, but they did not take security into consideration or just used the traditional and time-consuming encryption techniques.

Unfortunately, as far as we know, there is currently no efficient PPCBIR schemes can resist known-plaintext attack. And none of the them perform experimental tests over medical image databases, their accuracy is in terms of common database.

3 Models and Privacy Requirements

In this section, we formalize the system model, threat model, and clarify the privacy requirements.

3.1 System model

In the system model, we mainly focus on how to provide privacy-preserving similar medical image retrieval services with high accuracy. Each query user (e.g. physician) is equipped with a PC/smartphone and the image owner (e.g., medical institution) is equipped with a workstation, both PC/smartphone and workstation are used to pre-process and encrypt the original images, as well as communicate with others. Specifically, the system model of our work involves three main entities, namely Image Owner (IO), Query Users (QUs), and Cloud Server (CS), which are demonstrated in Fig. 1.

- IO: IO owns the medical image database, it needs to extract feature vectors and cluster similar ones. After that, IO encrypts the images and indexes (i.e., features vectors and cluster centers), and uploads them to CS. Besides, IO is also responsible for providing registration services to QUs.
- QUs $= \{\mathrm{QU}_1, \mathrm{QU}_2, \dots\}$: QUs is a collection of query users, and each QU$\in$ QUs needs to request keys by registering with IO at first, then generate trapdoor to request similar images search service from CS. After receiving the final result, QU can decrypt it to read the original images.
- CS: CS is assumed to have abundant storage space and powerful computing ability, which is also regarded as a link between IO and QUs. For one thing, it can storage encrypted data uploaded by IO. For another, it can perform calculations over stored data to provide search services for QUs.

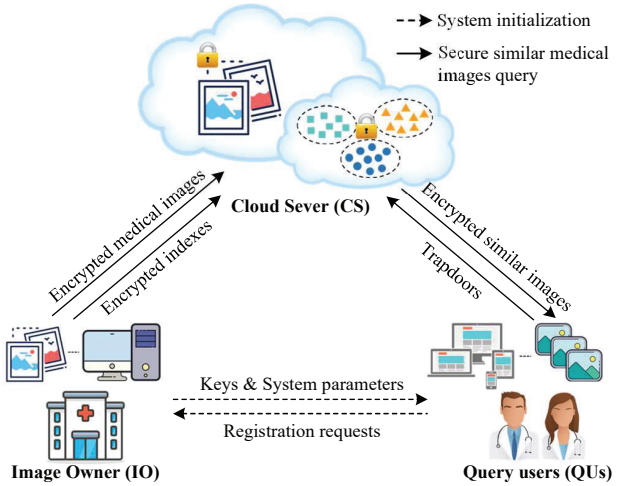


Fig. 1. System model under consideration

3.2 Threat model

In our system model, we assume that IO and QUs are trustable, IO will honestly share keys with registered QUs and provide encrypted data to CS, as well as QUs will send correct encrypted queries to CS. CS is considered to be *honest-but-curious* [32], that means, CS will follow the protocol strictly to store the outsourced data from IO and offer images retrieval services to QUs, but it may try to analyze encrypted data and queries due to its interest in original information.

Based on the knowledge CS or adversary possesses, we consider the Know-Plaintext Attack (KPA) model [33], [34], whose attack capability is defined as

Definition 1. *Known-plaintext attack (KPA). On the basis of COA, the attacker is assumed to have the ability to achieve a set of tuples in images, indexes and queries, and she/he knows the corresponding ciphertexts of these tuples.*

COA and KBA attacks are widely considered in the threat model of image retrieval schemes [7], [12], [16]. But as far as we know, KPA, as the stronger attack than COA and KBA, has not been proven to be resisted in recent schemes similar to ours.

3.3 Privacy requirements

Under the above-mentioned system and threat models, to ensure the data privacy of each entity, the following privacy requirements should be met simultaneously.

1) *Images security.* Medical images database is the private property of IO, and it may contain patients' personal privacy, thus the content of the images which need to be outsourced should be kept secret from CS and unauthorized users.
2) *Indexes (queries) confidentiality.* The indexes (queries) are feature vectors extracted from medical images, they can reveal the characteristic of images, the proposed scheme must guarantee that the attacker cannot obtain the raw data from encrypted indexes and trapdoors (encrypted queries).

3) *Trapdoors unlinkability.* In each image retrieval process, the trapdoor is exposed to CS. In order to prevent sensitive information from being inferred based on the received trapdoors, the proposed scheme should ensure that CS can neither distinguish the difference nor deduce the relationship between trapdoors, even from the same query request.

TABLE 1
Definition of key notations

Notations	Definition
n	The dimension of feature vectors.
N	The number of samples/medical images.
$\vec{x}_i$, $\vec{y}$, $\vec{s}_i$, $\vec{q}$	n-dimensional feature vectors.
Σ	The covariance matrix of a dataset or a cluster.
$M_{\{1,2,3,4\}}$	$(n+2)\times(n+2)$ random invertible matrices.
$Tr(\cdot)$	The sum of diagonal entries in a matrix.
c	The number of clusters.
r, α_i, β, r_q	random numbers used in our matrix encryption technique.
$E(\cdot)$	The ciphertext consists of two parts.
$\vec{c}_{i/\tau}$	The cluster center of a cluster.
$Enc(\cdot)$	A symmetric encryption algorithm.
k_I	The encryption/decryption key of $Enc(\cdot)$.
$\Pi_{\tau,1\leq\tau\leq c}$	$(n+2)\times(n+2)$ random permutation matrices.
$\boldsymbol{\pi}$	A set of $(n+2)$ permutation vectors with $(n+2)$-dimensional.
$\lvert\cdot\rvert$	The number of query vectors in a cluster.
$\mathcal{R}_{n\times2}$, $\mathcal{R}_{2\times n}$, $\mathcal{R}_{2\times2}$	Three random matrices used to extend matrix.

4 PRELIMINARIES

In this section, we briefly review the Mahalanobis distance [35], and an MD based Fuzzy C-Means algorithm [36], both of which are the building blocks of our proposed scheme. Moreover, the key notations used in this paper are listed in TABLE 1.

4.1 Mahalanobis distance

Given a vector $\vec{y} = (y_1, y_2, \ldots, y_n) \in \mathbb{S}^n$ and a dataset $\boldsymbol{S} = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_i, \ldots, \vec{x}_N\}$, where each sample $\vec{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in}) \in \mathbb{S}^n$, the Mahalanobis distance (MD) between $\vec{y}$ and $\vec{x}_i$ can be calculated by the following equation

$$D_{ma}(\vec{y}, \vec{x}_i) = \sqrt{(\vec{y} - \vec{x}_i)\Sigma^{-1}(\vec{y} - \vec{x}_i)^T},$$

where Σ denotes the covariance matrix of $\boldsymbol{S}$, and each element $\Sigma_{(s,t)}$ in the covariance matrix represents the covariance of $\vec{d}_s$ and $\vec{d}_t$, $s, t \in [1, n]$. Given a vector $\vec{d}_j = (x_{1j}, x_{2j}, \ldots, x_{Nj})$, $j \in [1, n]$, assume that

$$\overline{d_j} = \frac{1}{N}(x_{1j} + x_{2j} + \cdots + x_{Nj}),$$

$\Sigma_{(s,t)}$ can be denoted as

$$\Sigma_{(s,t)} = \frac{1}{N-1}\sum_{i=1}^{N}(x_{is} - \overline{d_s})(x_{it} - \overline{d_t}).$$

4.2 An MD based Fuzzy C-Means algorithm

The MD based Fuzzy C-Means (FCM-M) algorithm [36], which adopts an iterative clustering method, is proposed for partitioning a finite dataset $\{x_1, x_2, \ldots, x_n\}$ into c fuzzy clusters based on the Mahalanobis distance function. Specifically, the objective function of FCM-M algorithm is given as follows:

$$J_{FCM-M} = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m \cdot D_{ma}^2(x_j, \vec{c}_i),$$

where $m \in [1, +\infty)$ is fuzzy control, u_{ij} are the elements which consist of the membership matrix $U_{c\times n}$. Specifically, u_{ij} denotes the membership value of x_j in i-th cluster, and $\vec{c}_i$ is the cluster center of the i-th cluster, where $i \in [1, c], j \in [1, n]$. To obtain the final c fuzzy clusters, the following operations should be executed.

At first, initialize the number of clusters c, exponent value m, and the threshold value ε.

Then, initialize $U_{c\times n}^{(0)}$, whose elements u_{ij} are random numerical value chosen in the range $[0, 1]$, and satisfy the condition $\sum_{1\leq i\leq c} u_{ij} = 1$, where $j = 1, 2, \ldots, n$.

Next, For each x_j, calculate its i-th cluster center $\vec{c}_i$, covariance matrix Σ_i and membership value u_{ij} in sequence,

$$\vec{c}_i = \left[\sum_{j=1}^{n}(u_{ij})^m\right]^{-1}\left[\sum_{j=1}^{n}(u_{ij})^m x_j\right],$$

$$\Sigma_i = \left[\sum_{j=1}^{n}(u_{ij})^m\right]^{-1}\left[\sum_{j=1}^{n}(u_{ij})^m (x_j - \vec{c}_i)(x_j - \vec{c}_i)'\right],$$

$$u_{ij} = \left[\sum_{s=1}^{c}\left[\frac{(x_j - \vec{c}_i)'(\Sigma_i^{-1})(x_j - \vec{c}_i) - \ln|\Sigma_i^{-1}|}{(x_j - \vec{c}_s)'(\Sigma_s^{-1})(x_j - \vec{c}_s) - \ln|\Sigma_s^{-1}|}\right]^{\frac{1}{m-1}}\right]^{-1}.$$

According to the value of u_{ij}, update $U_{c\times n}^{(0)}$ to $U_{c\times n}^{(1)}$, if $||U_{c\times n}^{(1)} - U_{c\times n}^{(0)}|| \leq \varepsilon$, return the c clusters, otherwise repeat the operation of the previous step until $||U_{c\times n}^{(k)} - U_{c\times n}^{(k-1)}|| \leq \varepsilon$ is established, where k is the number of repetitions.

At last, after k rounds of calculation, the final c fuzzy clusters can be achieved.

5 A PRIVACY-PRESERVING COMPARISON METHOD FOR MAHALANOBIS DISTANCE

In this section, for the purpose of achieving secure calculation and comparison of Mahalanobis distance, we propose a privacy-preserving Mahalanobis distance comparison method called PMDC, which based on an enhanced matrix encryption method [17] designed for comparing Euclidean distance securely. Specifically, assume there is a database of samples $\mathcal{D} = \{\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_N\}$, where $\vec{s}_i = (s_{i1}, s_{i2}, \ldots, s_{in}), i \in [1, N]$, and the covariance matrix of $\mathcal{D}$ is $\Sigma_{\mathcal{D}}$. To compare the distance between different samples (e.g., $\vec{s}_i$ and $\vec{s}_j$, $i \neq j, i, j \in [1, N]$) in $\mathcal{D}$ and a given query $\vec{q} = (q_1, q_2, \ldots, q_n)$, the following four algorithms contained in PMDC, namely **KeyGen**, **DataEnc**, **TrapGen**, **DisCompare**, should be performed.

KeyGen(n): Given the dimensions of samples n, two $(n+2)\times(n+2)$ invertible matrices M_1 and M_2 are randomly selected as the secret key $sk = \{M_1, M_2\}$.

Algorithm 1: PMDC_**DataEnc**

Input: Sample database $\mathcal{D} = \{\vec{s}_1, \ldots, \vec{s}_N\}$, $\mathcal{D}$'s covariance matrix $\Sigma_{\mathcal{D}}$, secret key $sk = \{M_1, M_2\}$.
Output: Encrypted samples $\{E(\vec{s}_1), \ldots, E(\vec{s}_N)\}$.

for $1 \le i \le N$ **do**
  $\mathcal{W}_i = \vec{s}_i \cdot \Sigma_{\mathcal{D}}^{-1} \cdot \vec{s}_i^T$;
  $\widetilde{\boldsymbol{s}}_i = \vec{s}_i \cdot (\Sigma_{\mathcal{D}}^{-1})^T$;
  $\overline{\boldsymbol{s}}_i = \vec{s}_i \cdot \Sigma_{\mathcal{D}}^{-1}$;
  Randomly choose a number α_i;
  $\widetilde{\boldsymbol{s}}_i[n] \leftarrow \alpha_i$, $\widetilde{\boldsymbol{s}}_i[n+1] \leftarrow 1$;
  $\overline{\boldsymbol{s}}_i[n] \leftarrow \mathcal{W}_i$, $\overline{\boldsymbol{s}}_i[n+1] \leftarrow -\alpha_i$;
  Initialize two $(n+2) \times (n+2)$ matrices $\widetilde{S}_i$, $\overline{S}_i$;
  for $0 \le x \le n+1, 0 \le y \le n+1$ **do**
    if $x == y$ **then**
      $\widetilde{S}_i[x][y] = \widetilde{\boldsymbol{s}}_i[x]$, $\overline{S}_i[x][y] = \overline{\boldsymbol{s}}_i[x]$;
    else $\widetilde{S}_i[x][y] = 0$, $\overline{S}_i[x][y] = 0$;
  Randomly choose two lower triangular matrices $\widetilde{Q}_i$, $\overline{Q}_i$ with diagonal entries set to 1;
  $E(\widetilde{\boldsymbol{s}}_i) = M_1 \widetilde{Q}_i \widetilde{S}_i M_2$;
  $E(\overline{\boldsymbol{s}}_i) = M_1 \overline{Q}_i \overline{S}_i M_2$;
  $E(\vec{s}_i) \leftarrow \{E(\widetilde{\boldsymbol{s}}_i), E(\overline{\boldsymbol{s}}_i)\}$;
return Encrypted samples $\{E(\vec{s}_1), E(\vec{s}_2), \ldots, E(\vec{s}_N)\}$.

Algorithm 2: PMDC_**TrapGen**

Input: Query vector $\vec{q} = (q_1, q_2, \ldots, q_n)$, secret keys $sk = \{M_1, M_2\}$.
Output: Trapdoor $E(\vec{q})$.

Initialize two vectors $\widetilde{\boldsymbol{q}} = \overline{\boldsymbol{q}} = -\vec{q}$;
Randomly choose two number β and r_q;
$\widetilde{\boldsymbol{q}}[n] \leftarrow \beta$, $\widetilde{\boldsymbol{q}}[n+1] \leftarrow r_q$;
$\overline{\boldsymbol{q}}[n] \leftarrow 1$, $\overline{\boldsymbol{q}}[n+1] \leftarrow \beta$;
Initialize two $(n+2) \times (n+2)$ matrices $\widetilde{\mathcal{Q}}$, $\overline{\mathcal{Q}}$;
for $0 \le x \le n+1, 0 \le y \le n+1$ **do**
  if $x = y$ **then**
    $\widetilde{\mathcal{Q}}[x][y] = \widetilde{\boldsymbol{q}}[x]$, $\overline{\mathcal{Q}}[x][y] = \overline{\boldsymbol{q}}[x]$;
  else $\widetilde{\mathcal{Q}}[x][y] = 0$, $\overline{\mathcal{Q}}[x][y] = 0$;
Randomly choose two lower triangular matrices $\widetilde{Q}_\omega$, $\overline{Q}_\omega$ with diagonal entries set to 1;
$E(\widetilde{\boldsymbol{q}}) = M_2^{-1} \widetilde{\mathcal{Q}} \widetilde{Q} M_1^{-1}$;
$E(\overline{\boldsymbol{q}}) = M_2^{-1} \overline{\mathcal{Q}}\, \overline{Q} M_1^{-1}$;
$E(\vec{q}) \leftarrow \{E(\widetilde{\boldsymbol{q}}), E(\overline{\boldsymbol{q}})\}$;
return Trapdoor $E(\vec{q})$.

Algorithm 3: PMDC_**DisCompare**

Input: Encrypted samples $\{E(\vec{s}_1), E(\vec{s}_2), \ldots, E(\vec{s}_N)\}$, the trapdoor $E(\vec{q})$.
Output: A collection AD with ascending sorting.

for $1 \le i \le N$ **do**
  $\widetilde{P}_i = E(\widetilde{\boldsymbol{s}}_i) \cdot E(\widetilde{\boldsymbol{q}})$;
  $\overline{P}_i = E(\overline{\boldsymbol{s}}_i) \cdot E(\overline{\boldsymbol{q}})$;
  $D_i = Tr(\widetilde{P}_i) + Tr(\overline{P}_i)$;
Sort $\{D_1, D_2, \ldots, D_N\}$ in ascending order to form a new collection AD;
return AD.

DataEnc($\mathcal{D}$, $\Sigma_{\mathcal{D}}$, sk): Given database $\mathcal{D}$, $\mathcal{D}$'s covariance matrix $\Sigma_{\mathcal{D}}$ and secret key sk, this algorithm outputs the encrypted samples $\{E(\vec{s}_1), \ldots, E(\vec{s}_N)\}$ as shown in **Algorithm** 1. For each $\vec{s}_i$ in $\mathcal{D}$, the encryptor first calculates $\mathcal{W}_i = \vec{s}_i \cdot \Sigma_{\mathcal{D}}^{-1} \cdot \vec{s}_i^T$, $\widetilde{\boldsymbol{s}}_i = \vec{s}_i \cdot (\Sigma_{\mathcal{D}}^{-1})^T$ and $\overline{\boldsymbol{s}}_i = \vec{s}_i \cdot \Sigma_{\mathcal{D}}^{-1}$ in respective, where $\widetilde{\boldsymbol{s}}_i$ and $\overline{\boldsymbol{s}}_i$ are n-dimensional vectors, denoted as $\widetilde{\boldsymbol{s}}_i = (\widetilde{s}_{i1}, \widetilde{s}_{i2}, \ldots, \widetilde{s}_{in})$, $\overline{\boldsymbol{s}}_i = (\overline{s}_{i1}, \overline{s}_{i2}, \ldots, \overline{s}_{in})$. Then, $\widetilde{\boldsymbol{s}}_i$ is extended into an $(n+2)$-dimensional vector $\widetilde{\boldsymbol{s}}_i = (\widetilde{s}_{i1}, \ldots, \widetilde{s}_{in}, \alpha_i, 1)$, where $\widetilde{s}_{i(n+1)}$ is a random number α_i and $\widetilde{s}_{i(n+2)}$ is set to 1. Meanwhile, $\overline{\boldsymbol{s}}_i$ is extended into $\overline{\boldsymbol{s}}_i = (\overline{s}_{i1}, \ldots, \overline{s}_{in}, \mathcal{W}_i, -\alpha_i)$, where $\overline{s}_{i(n+1)} = \mathcal{W}_i$ and $\overline{s}_{i(n+2)} = -\alpha_i$. After that, two diagonal matrices $\widetilde{S}_i$ and $\overline{S}_i$, whose diagonal entries are the elements in $\widetilde{\boldsymbol{s}}_i$ and $\overline{\boldsymbol{s}}_i$ in respective, are constructed. Finally, each sample $\vec{s}_i$ is encrypted into $E(\vec{s}_i) = \{E(\widetilde{\boldsymbol{s}}_i), E(\overline{\boldsymbol{s}}_i)\}$ by $E(\widetilde{\boldsymbol{s}}_i) = M_1 \widetilde{Q}_i \widetilde{S}_i M_2$, $E(\overline{\boldsymbol{s}}_i) = M_1 \overline{Q}_i \overline{S}_i M_2$, where $\widetilde{Q}_i$, $\overline{Q}_i$ are two random $(n+2) \times (n+2)$ lower triangular matrices whose diagonal entries set to 1.

TrapGen($\vec{q}$, sk): Given a query vector $\vec{q}$ and the secret key sk, this algorithm outputs a trapdoor $E(\vec{q})$ as shown in **Algorithm** 2. Similar to **DataEnc**, the generator first extends $\vec{q}$ to $\widetilde{\boldsymbol{q}}$ and $\overline{\boldsymbol{q}}$, where $\widetilde{\boldsymbol{q}} = (-q_1, \ldots, -q_n, \beta, r_q)$, $\overline{\boldsymbol{q}} = (-q_1, \ldots, -q_n, 1, \beta)$, β and r_q are two random numbers. Then, the elements in $\widetilde{\boldsymbol{q}}$ and $\overline{\boldsymbol{q}}$ are taken as diagonal entries to construct diagonal matrices $\widetilde{\mathcal{Q}}$ and $\overline{\mathcal{Q}}$, respectively. After that, the generator randomly generates two $(n+2) \times (n+2)$ lower triangular matrices $\widetilde{Q}$ and $\overline{Q}$ whose diagonal entries are set to 1. Finally, $\vec{q}$ is encrypted into $E(\vec{q}) = \{E(\widetilde{\boldsymbol{q}}), E(\overline{\boldsymbol{q}})\}$ by $E(\widetilde{\boldsymbol{q}}) = M_2^{-1} \widetilde{\mathcal{Q}} \widetilde{Q} M_1^{-1}$ and $E(\overline{\boldsymbol{q}}) = M_2^{-1} \overline{\mathcal{Q}}\, \overline{Q} M_1^{-1}$.

DisCompare($\{E(\vec{s}_1), E(\vec{s}_2), \ldots, E(\vec{s}_N)\}$, $E(\vec{q})$): Given encrypted samples $\{E(\vec{s}_1), E(\vec{s}_2), \ldots, E(\vec{s}_N)\}$, and the trapdoor $E(\vec{q})$, this algorithm outputs a collection AD with ascending order as shown in **Algorithm** 3. For each $E(\vec{s}_i) = \{E(\widetilde{\boldsymbol{s}}_i), E(\overline{\boldsymbol{s}}_i)\}$, the comparator first computes two matrices $\widetilde{P}_i = E(\widetilde{\boldsymbol{s}}_i) \cdot E(\widetilde{\boldsymbol{q}})$, $\overline{P}_i = E(\overline{\boldsymbol{s}}_i) \cdot E(\overline{\boldsymbol{q}})$. Then, it calculates $D_i = Tr(\widetilde{P}_i) + Tr(\overline{P}_i)$, where $Tr(\cdot)$ denotes the trace of a matrix (i.e., the sum of a matrix's diagonal entries). Finally, $\{D_1, D_2, \ldots, D_N\}$ are sorted in ascending order and form a new collection AD, which can map the distances from the samples to the query.

Correctness of PMDC. According to the above four algorithms, it is obvious that the correctness of PMDC depends on the establishment of $D_i \le D_j \Rightarrow D_{ma}(\vec{q}, \vec{s}_i) \le D_{ma}(\vec{q}, \vec{s}_j)$, where $i, j \in [1, N]$.

Proof. Due to $D_i = Tr(\widetilde{P}_i) + Tr(\overline{P}_i)$, we first expand $\widetilde{P}_i$ and $\overline{P}_i$ as

$$\begin{aligned}
\widetilde{P}_i &= E(\widetilde{\boldsymbol{s}}_i) \cdot E(\widetilde{\boldsymbol{q}}) \\
&= M_1 \widetilde{Q}_i \widetilde{S}_i M_2 \cdot M_2^{-1} \widetilde{\mathcal{Q}} \widetilde{Q} M_1^{-1} \\
&= M_1 \widetilde{Q}_i \widetilde{S}_i \widetilde{\mathcal{Q}} \widetilde{Q} M_1^{-1}, \\
\overline{P}_i &= E(\overline{\boldsymbol{s}}_i) \cdot E(\overline{\boldsymbol{q}}) \\
&= M_1 \overline{Q}_i \overline{S}_i M_2 \cdot M_2^{-1} \overline{\mathcal{Q}\mathcal{Q}} M_1^{-1} \\
&= M_1 \overline{Q}_i \overline{S}_i \overline{\mathcal{Q}}\, \overline{Q} M_1^{-1}.
\end{aligned}$$

According to the property of *similar matrices*, we can observe that the traces of similarly transformed matrices are similarity-invariant, which means

$$Tr(\widetilde{P}_i) = Tr(\widetilde{Q}_i \widetilde{S}_i \widetilde{\mathcal{Q}} \widetilde{Q}),$$

$$Tr(\overline{P}_i) = Tr(\overline{Q}_i \overline{S}_i \overline{\mathcal{Q}}\, \overline{Q}).$$

Besides, both $\widetilde{Q}_i$ and $\widetilde{Q}$ are random lower triangular matrices whose diagonal entries are set to 1, thus $\widetilde{Q}_i\widetilde{S}_i\widetilde{\mathcal{Q}}\widetilde{Q}$ can be represented as

$$\begin{bmatrix} -\widetilde{s}_{i1}q_1 & 0 & \cdots & 0 & 0 \\ \widetilde{r}_{21} & -\widetilde{s}_{i2}q_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \widetilde{r}_{(n+1)1} & \widetilde{r}_{(n+1)2} & \cdots & \alpha_i\beta & 0 \\ \widetilde{r}_{(n+2)1} & \widetilde{r}_{(n+2)2} & \cdots & \widetilde{r}_{(n+2)(n+1)} & r_q \end{bmatrix},$$

where $\widetilde{r}_{ab}, a \in [2, n+2], b \in [1, n+1]$ are random numbers calculated by the elements from $\widetilde{Q}_i$, $\widetilde{Q}$, $\widetilde{s}_i$ and $\vec{q}$. Obviously,

$$Tr(\widetilde{Q}_i\widetilde{S}_i\widetilde{\mathcal{Q}}\widetilde{Q}) = Tr(\widetilde{S}_i\widetilde{\mathcal{Q}}).$$

Similar to $\widetilde{Q}_i\widetilde{S}_i\widetilde{\mathcal{Q}}\widetilde{Q}$, we have

$$Tr(\overline{Q}_i\overline{S}_i\overline{\mathcal{Q}}\,\overline{Q}) = Tr(\overline{S}_i\overline{\mathcal{Q}}).$$

In summary, it can be inferred that the traces of $\widetilde{P}_i$ ($\overline{P}_i$) and $\widetilde{S}_i\widetilde{\mathcal{Q}}$ ($\overline{S}_i\overline{\mathcal{Q}}$) are equal, thus

$$Tr(\widetilde{P}_i) = Tr(\widetilde{S}_i\widetilde{\mathcal{Q}}) = \alpha_i\beta + r_q - \sum_{k=1}^{n}\widetilde{s}_{ik}q_k,$$
$$Tr(\overline{P}_i) = Tr(\overline{S}_i\overline{\mathcal{Q}}) = \mathcal{W}_i - \alpha_i\beta - \sum_{k=1}^{n}\overline{s}_{ik}q_k.$$

Then, the value of D_i can be obtained by

$$\begin{aligned} D_i &= Tr(\widetilde{P}_i) + Tr(\overline{P}_i) \\ &= \mathcal{W}_i - \sum_{k=1}^{n}\widetilde{s}_{ik}q_k - \sum_{k=1}^{n}\overline{s}_{ik}q_k + r_q. \end{aligned}$$

Assume there exists D_i and D_j, $i \neq j$, $i, j \in [1, N]$, the difference of D_i and D_j can be calculated as follows

$$\begin{aligned} D_i - D_j =&(\mathcal{W}_i - \sum_{k=1}^{n}\widetilde{s}_{ik}q_k - \sum_{k=1}^{n}\overline{s}_{ik}q_k + r_q) \\ &-(\mathcal{W}_j - \sum_{k=1}^{n}\widetilde{s}_{jk}q_k - \sum_{k=1}^{n}\overline{s}_{jk}q_k + r_q) \\ =&(\mathcal{W}_i - \sum_{k=1}^{n}\widetilde{s}_{ik}q_k - \sum_{k=1}^{n}\overline{s}_{ik}q_k) \\ &-(\mathcal{W}_j - \sum_{k=1}^{n}\widetilde{s}_{jk}q_k - \sum_{k=1}^{n}\overline{s}_{jk}q_k). \end{aligned}$$

Meanwhile, according to the rules of matrix multiplication, the following equations are valid

$$[\vec{s}_{\{i,j\}}(\Sigma_{\mathcal{D}}^{-1})^T] \cdot \vec{q}^T = \sum_{k=1}^{n}\widetilde{s}_{\{i,j\}k}q_k,$$

$$[\vec{s}_{\{i,j\}}\Sigma_{\mathcal{D}}^{-1}] \cdot \vec{q}^T = \sum_{k=1}^{n}\overline{s}_{\{i,j\}k}q_k,$$

$$\vec{s}_{\{i,j\}}(\Sigma_{\mathcal{D}}^{-1})^T\vec{q}^T = (\vec{q}\,\Sigma_{\mathcal{D}}^{-1}\vec{s}_{\{i,j\}}^T)^T = \vec{q}\,\Sigma_{\mathcal{D}}^{-1}\vec{s}_{\{i,j\}}^T.$$

Therefore, with $\mathcal{W}_{\{i,j\}} = \vec{s}_{\{i,j\}} \cdot \Sigma_{\mathcal{D}}^{-1} \cdot \vec{s}_{\{i,j\}}^T$, we have

$$\begin{aligned} D_i - D_j =& \left[\vec{s}_i\Sigma_{\mathcal{D}}^{-1}\vec{s}_i^T - \vec{s}_i(\Sigma_{\mathcal{D}}^{-1})^T\vec{q}^T - \vec{s}_i\Sigma_{\mathcal{D}}^{-1}\vec{q}^T\right] \\ &-\left[\vec{s}_j\Sigma_{\mathcal{D}}^{-1}\vec{s}_j^T - \vec{s}_j(\Sigma_{\mathcal{D}}^{-1})^T\vec{q}^T - \vec{s}_j\Sigma_{\mathcal{D}}^{-1}\vec{q}^T\right] \\ =&\left[\vec{q}\,\Sigma_{\mathcal{D}}^{-1}\vec{q}^T - \vec{q}\,\Sigma_{\mathcal{D}}^{-1}\vec{s}_i^T - \vec{s}_i\Sigma_{\mathcal{D}}^{-1}\vec{q}^T + \vec{s}_i\Sigma_{\mathcal{D}}^{-1}\vec{s}_i^T\right] \\ &-\left[\vec{q}\,\Sigma_{\mathcal{D}}^{-1}\vec{q}^T - \vec{q}\,\Sigma_{\mathcal{D}}^{-1}\vec{s}_j^T - \vec{s}_j\Sigma_{\mathcal{D}}^{-1}\vec{q}^T + \vec{s}_j\Sigma_{\mathcal{D}}^{-1}\vec{s}_j^T\right] \\ =&(\vec{q}-\vec{s}_i)\Sigma_{\mathcal{D}}^{-1}(\vec{q}-\vec{s}_i)^T - (\vec{q}-\vec{s}_j)\Sigma_{\mathcal{D}}^{-1}(\vec{q}-\vec{s}_j)^T \\ =&D_{ma}^2(\vec{q},\vec{s}_i) - D_{ma}^2(\vec{q},\vec{s}_j). \end{aligned}$$

Since $D_{ma}(\vec{q}, \vec{s}_{\{i,j\}}) > 0$, the correctness of $D_i \leq D_j \Rightarrow D_{ma}(\vec{q},\vec{s}_i) \leq D_{ma}(\vec{q},\vec{s}_j)$ can be verified easily. □

6 Accurate and Privacy-Preserving Similar Medical Image Retrieval Scheme

In this section, an accurate and privacy-preserving similar medical image retrieval scheme, named TAMMIE, is proposed based on PMDC. TAMMIE mainly consists of four phases: *1) system initialization*, *2) encrypted images and clusters outsourcing*, *3) trapdoor generation* and *4) similar images retrieval and reading*. The overview of TAMMIE is described in Fig. 2. At first, IO extracts and compresses feature vectors for all medical images stored in it by employing a pre-trained Convolutional Neural Network (CNN) model and Principal Component Analysis (PCA) [37] technique. Then, to improve the efficiency of retrieval, FCM-M clustering algorithm is utilized to partition feature vectors into c clusters. Next, according to the dimension of feature vectors, four invertible matrices and c permutation matrices are generated as secret keys and shared with QUs. After that, the clusters and original images would be encrypted with different cryptographic primitives and sent to CS. Besides, CS will receive trapdoors from bulk QUs, it is responsible for CS to find the most similar cluster and the top-k similar images according to the trapdoors. Finally, QUs can receive the encrypted similar images sent by CS and decrypt the images to read the final result.

6.1 System initialization

In this phase, IO first extracts and clusters the medical images' feature vectors. Then it chooses a symmetric encryption algorithm and generates system parameters. Besides, this phase also provides QUs with registration services.

At first, based on a pre-trained CNN model, which discards the last output layer compared with a typical CNN [38], IO extracts feature vectors from the original images in the medical image database MI= $\{\mathrm{I}_1, \mathrm{I}_2, \ldots, \mathrm{I}_N\}$. To compress the dimension of the extracted feature vectors, Principal Component Analysis (PCA) technique is employed to generate low-dimensional vector $\vec{s}_i$ for I_i, $i \in [1, N]$, where $\vec{s}_i = \{s_{i1}, s_{i2}, \ldots, s_{in}\}$.

Then, to speed up the retrieval of similar images, FCM-M clustering algorithm is introduced in our proposed scheme, with the objective function

$$J_{FCM-M} = \sum_{\tau=1}^{c}\sum_{i=1}^{N} u_{\tau i}^m \cdot D_{ma}^2(\vec{s}_i, \vec{c}_\tau).$$

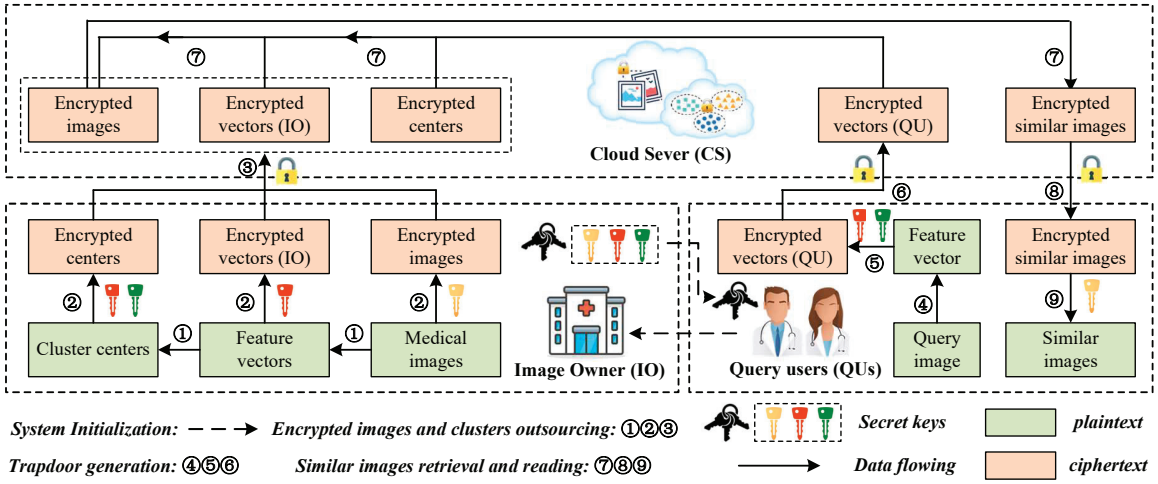


Fig. 2. Overview of TAMMIE

Algorithm 4: Cluster centers encryption

Input: Cluster centers with their corresponding covariance matrices $(\vec{c}_\tau, \Sigma_\tau)$, random permutation vectors π_γ, $1 \le \gamma \le n+2$, permutation matrices Π_τ, $1 \le \tau \le c$, secret keys $sk_1 = \{M_1, M_2\}$ and $sk_2 = \{M_3, M_4\}$.

Output: Encrypted cluster centers $\{\mathrm{EC}_1, \ldots, \mathrm{EC}_c\}$.

for $1 \le \tau \le c$ **do**
- $E(\vec{c}_\tau)$=PMDC_**DataEnc**$(\vec{c}_\tau, \Sigma_\tau, k_V)$;
- Initialize an $(n+2) \times (n+2)$ empty matrix Σ'_τ;
- **for** $0 \le i \le n+1, 0 \le j \le n+1$ **do**
  - **if** $i < n$ && $j < n$ **then** $\Sigma'_\tau[i][j] = \Sigma_\tau^{-1}[i][j]$;
  - **else** $\Sigma'_\tau[i][j] = Randomnumber()$;
- Initialize an $(n+2) \times (n+2)$ empty matrix Σ''_τ;
- Initialize an $(n+2)$-dimensional array $Column$;
- **for** $0 \le j \le n+1$ **do**
  - **for** $0 \le i \le n+1$ **do** $Column[i] = \Sigma'_\tau[i][j]$;
  - $Column = \pi_{j+1}(Column)$;
  - **for** $0 \le i \le n+1$ **do** $\Sigma''_\tau[i][j] = Column[i]$;
- $E(\Sigma_\tau) = M_3 \Sigma''_\tau \Pi_\tau^T M_4$;
- $\mathrm{EC}_\tau \leftarrow \{E(\vec{c}_\tau), E(\Sigma_\tau)\}$;

return Encrypted cluster centers $\{\mathrm{EC}_1, \ldots, \mathrm{EC}_c\}$.

Thus, all feature vectors $\{\vec{s}_1, \ldots, \vec{s}_N\}$ can be divided into c clusters $\{\mathcal{C}_1, \ldots, \mathcal{C}_c\}$ with cluster centers $\{\vec{c}_1, \ldots, \vec{c}_c\}$ and covariance matrices $\{\Sigma_1, \ldots, \Sigma_c\}$, the cluster center $\vec{c}_\tau = (c_{\tau 1}, \ldots, c_{\tau n})$, $\tau \in [1, c]$.

Next, IO randomly chooses a symmetric encryption algorithm $Enc(\cdot)$ and generates the corresponding secret key k_I. Besides, IO also randomly chooses four invertible matrices M_1, M_2, M_3, M_4, c permutation matrices $\mathbf{\Pi} = \{\Pi_1, \ldots, \Pi_c\}$ and $n+2$ permutation vectors $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_{n+2}\}$, where the $c+4$ matrices are with $(n+2)$ rows and $(n+2)$ columns, and each permutation vector is $(n+2)$-dimensional.

When registering in IO, QUs submit their identification and related information to IO via a secure channel. If QU is considered to be a legal user, she/he will receive the collection SP= $\{k_I, M_1, M_2, M_3, M_4, \mathbf{\Pi}, \boldsymbol{\pi}\}$ from IO; otherwise the registration fails, and CS would not provide correct query result to users who fail to register.

6.2 Encrypted images and indexes outsourcing

In this phase, IO encrypts outsourced images, extracted features vectors, and c cluster centers. And sending all of them to CS to provide similar image retrieval service.

• Stage 1. Original images Encryption

Given a medical image database MI= $\{\mathrm{I}_1, \mathrm{I}_2, \ldots, \mathrm{I}_N\}$, IO encrypts MI to EI with its secret key k_I as

$$\begin{aligned}\mathrm{EI} &= Enc(\mathrm{MI}, k_I) \\ &= \{Enc(\mathrm{I}_1, k_I), Enc(\mathrm{I}_2, k_I), \ldots, Enc(\mathrm{I}_N, k_I)\} \\ &= \{\mathrm{E}_1, \mathrm{E}_2, \ldots, \mathrm{E}_N\}.\end{aligned}$$

• Stage 2. Feature vectors Encryption

All extracted feature vectors have been partitioned into c clusters, for each cluster $\mathcal{C}_\tau$ with the covariance matrix Σ_τ $(1 \le \tau \le c)$, IO directly encrypts all feature vectors in the cluster by running PMDC_**DataEnc**$(\mathcal{C}_\tau, \Sigma_\tau, sk_1)$, where $sk_1 = \{M_1, M_2\}$. As a result, IO can obtain a set $\mathrm{EV}_\tau = \{E(\vec{s}_{\tau 1}), \ldots, E(\vec{s}_{\tau\kappa}), \ldots, E(\vec{s}_{\tau l_\tau})\}$, where $E(\vec{s}_{\tau\kappa}) = \{E(\tilde{\boldsymbol{s}}_{\tau\kappa}), E(\overline{\boldsymbol{s}}_{\tau\kappa})\}$, $l_\tau = |\mathcal{C}_\tau|$, $\sum_{\tau=1}^{c} l_\tau = N$.

• Stage 3. Cluster centers Encryption

Due to different cluster centers with different covariance matrices, the distance secure comparison between query vector and cluster centers cannot be executed only by utilizing PMDC_**DataEnc**. Thus, as shown in **Algorithm** 4, for each cluster center $\vec{c}_\tau = (c_{\tau 1}, c_{\tau 2}, \ldots, c_{\tau n})$, IO firstly employs PMDC_**DataEnc** to encrypt $\vec{c}_\tau$ into $E(\vec{c}_\tau) = \{E(\tilde{\boldsymbol{c}}_\tau), E(\overline{\boldsymbol{c}}_\tau)\}$ with sk_1. Then, it calculates the inverse of Σ_τ and extend Σ_τ^{-1} into a $(n+2) \times (n+2)$ matrix Σ'_τ by inserting random numbers:

$$\Sigma_\tau^{-1} \mapsto \Sigma'_\tau = \begin{bmatrix} \Sigma_\tau^{-1} & \mathcal{R}_{n\times 2}^{(\tau 1)} \\ \mathcal{R}_{2\times n}^{(\tau 2)} & \mathcal{R}_{2\times 2}^{(\tau 3)} \end{bmatrix},$$

where $\mathcal{R}_{n\times 2}^{(\tau 1)}$, $\mathcal{R}_{2\times n}^{(\tau 2)}$, $\mathcal{R}_{2\times 2}^{(\tau 3)}$ are three random matrices chosen by IO. And for the γ-th column $\Sigma'_\tau(\gamma_{col})$ in Σ'_τ, the permutation vector π_γ, $\gamma \in [1, n+2]$ is used to change the position of the elements in $\Sigma'_\tau(\gamma_{col})$, thus a new matrix Σ''_τ can be constructed as

$$\Sigma''_\tau = [\pi_1 \left(\Sigma'_\tau(1_{col})^T\right)^T, \ldots, \pi_{n+2} \left(\Sigma'_\tau((n+2)_{col})^T\right)^T].$$

After that, based on the τ-th random permutation matrix Π_τ and $sk_2 = \{M_3, M_4\}$, IO encrypts Σ_τ by executing

$$E(\Sigma_\tau) = M_3 \Sigma''_\tau \Pi_\tau^T M_4,$$

and denote the ciphertext of cluster center $\vec{c}_\tau$ as $\mathrm{EC}_\tau = \{E(\vec{c}_\tau), E(\Sigma_\tau)\}$.

Finally, all of the above-mentioned encrypted images, feature vectors and cluster centers $\{\mathrm{EI}, \mathrm{EV}_\tau|_{\tau=1}^c, \mathrm{EC}_\tau|_{\tau=1}^c\}$ are outsourced to CS.

6.3 Trapdoor generation

In this phase, each registered QU encrypts her/his original image's feature vector and requests similar image retrieval service from CS.

After successfully registering in IO, QU first extracts feature vector $\vec{q} = (q_1, q_2, \ldots, q_n)$ from her/his query image, then multiplies $\vec{q}$ by each q_j to obtain

$$\vec{\varphi}_j = q_j \cdot \vec{q} = q_j \cdot (q_1, \ldots, q_n) = (\varphi_{j1}, \ldots, \varphi_{jn}),$$

and constructs a $n \times n$ matrix

$$\Psi = \begin{bmatrix} \vec{\varphi}_1 \\ \vec{\varphi}_2 \\ \vdots \\ \vec{\varphi}_n \end{bmatrix} = \begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1n} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{n1} & \varphi_{n2} & \cdots & \varphi_{nn} \end{bmatrix}.$$

After that, IO extends Ψ to a $(n+2) \times (n+2)$ matrix as

$$\Psi \mapsto \Psi' = \begin{bmatrix} \Psi & \mathcal{R}_{n\times 2}^{(1)} \\ \mathcal{R}_{2\times n}^{(2)} & \mathcal{R}_{2\times 2}^{(3)} \end{bmatrix},$$

where $\mathcal{R}_{n\times 2}^{(1)}, \mathcal{R}_{2\times n}^{(2)}, \mathcal{R}_{2\times 2}^{(3)}$ are three random matrices chosen by QU.

Next, QU runs PMDC_**TrapGen**($\vec{q}$, sk_1) to encrypt $\vec{q}$ and obtains trapdoor $E(\vec{q}) = \{E(\widetilde{\boldsymbol{q}}), E(\overline{\boldsymbol{q}})\}$. Besides, to calculate the mahalanobis distance between $\vec{q}$ and cluster centers $\{\vec{c}_1, \ldots, \vec{c}_c\}$ with different covariance matrices, QU also needs to encrypt Ψ with $\boldsymbol{\pi}$, sk_2 and $\boldsymbol{\Pi}$. Specifically, IO first denotes the γ-th row of Ψ' as $\Psi'(\gamma_{row})$ and permutes it through the permutation vectors in $\boldsymbol{\pi}$ as

$$\Psi' \to \Psi'' = \begin{bmatrix} \pi_1(\Psi'(1_{row})) \\ \vdots \\ \pi_{n+2}(\Psi'((n+2)_{row})) \end{bmatrix},$$

then encrypts Ψ into $E_\tau(\Psi) = M_4^{-1} \Pi_\tau \Psi'' M_3^{-1}, 1 \le \tau \le c$ via different permutation matrices in $\boldsymbol{\Pi}$.

At last, QU sends the trapdoor $\mathrm{TD} = \{E(\vec{q}), E_\tau(\Psi)|_{\tau=1}^c\}$ to CS for similar image retrieval service.

6.4 Similar images retrieval and reading

In this phase, CS firtly finds the closest cluster to the query vector, then calculates the top-k similar images in the cluster and returns them to QU.

Upon receiving the trapdoor TD from QU, with the encrypted cluster center $\mathrm{EC}_\tau = \{E(\widetilde{\boldsymbol{c}}_\tau), E(\overline{\boldsymbol{c}}_\tau), E(\Sigma_\tau)\}$ for $1 \le \tau \le c$, CS firstly computes $\widetilde{P}_\tau = E(\widetilde{\boldsymbol{c}}_\tau) \cdot E(\widetilde{\boldsymbol{q}})$, $\overline{P}_\tau = E(\overline{\boldsymbol{c}}_\tau) \cdot E(\overline{\boldsymbol{q}})$ and $P_\tau = E_\tau(\Psi) \cdot E(\Sigma_\tau)$, then adds up the traces of $\widetilde{P}_\tau$, $\overline{P}_\tau$ and P_τ, $D_\tau = Tr(\widetilde{P}_\tau) + Tr(\overline{P}_\tau) + Tr(P_\tau)$. According to the value of $\{D_1, D_2, \ldots, D_c\}$, CS finds the smallest value $D_\eta = \min\{D_1, D_2, \ldots, D_c\}$, $\eta \in [1, c]$.

Next, CS extracts the η-th cluster EV_η as input to run PMDC_**DisCompare**($\{E(\vec{s}_{\eta 1}), \ldots, E(\vec{s}_{\eta l_\eta})\}, E(\vec{q})$). And CS can obtain a collection AD with ascending sorting. After that, the k encrypted images stored in EI, which corresponds to the top-k values in AD, are selected and returned to QU.

Finally, with the secret key k_I, QU decrypts and reads the received encrypted medical images.

Correctness of TAMMIE. It is clear that the correctness of TAMMIE depends on two aspects: *1) the correctness of* PMDC; *2) the correctness of* $D_{\tau_1} \le D_{\tau_2} \Rightarrow D_{ma}(\vec{q}, \vec{c}_{\tau_1}) \le D_{ma}(\vec{q}, \vec{c}_{\tau_2})$, where $\tau_1, \tau_2 \in \{1, 2, \ldots, c\}$, $\tau_1 \ne \tau_2$. Due to *1)* has been proved in Section 5, in this section, we are devoted to completing the proof of *2)*. The detailed derivation process is given as follows, and for ease of description, we use τ as a universal character to represent τ_1 or τ_2.

Proof. Firstly, we expand $\widetilde{P}_\tau$, $\overline{P}_\tau$ and P_τ as

$$\begin{aligned} \widetilde{P}_\tau &= E(\widetilde{\boldsymbol{c}}_\tau) \cdot E(\widetilde{\boldsymbol{q}}) = M_1 \widetilde{Q}_\tau \widetilde{C}_\tau \widetilde{\mathcal{Q}} \widetilde{Q} M_1^{-1}, \\ \overline{P}_\tau &= E(\overline{\boldsymbol{c}}_\tau) \cdot E(\overline{\boldsymbol{q}}) = M_1 \overline{Q}_\tau \overline{C}_\tau \overline{\mathcal{Q}} \overline{Q} M_1^{-1}, \\ P_\tau &= E_\tau(\Psi) \cdot E(\Sigma_\tau) = M_4^{-1} \Pi_\tau \Psi'' \Sigma''_\tau \Pi_\tau^T M_4, \end{aligned}$$

where $\widetilde{C}_\tau$, $\overline{C}_\tau$, $\widetilde{\mathcal{Q}}$ and $\overline{\mathcal{Q}}$ are four diagonal matrices and $\widetilde{Q}_\tau$, $\overline{Q}_\tau$, $\widetilde{Q}$ and $\overline{Q}$ are four lower triangular matrices with diagonal entries set to 1. According to the derivation of the correctness of PMDC, the following equations is satisfied:

$$\begin{aligned} Tr(\widetilde{P}_\tau) = Tr(\widetilde{C}_\tau \widetilde{\mathcal{Q}}),\ Tr(\overline{P}_\tau) = Tr(\overline{C}_\tau \overline{\mathcal{Q}}), \\ Tr(\widetilde{P}_\tau) + Tr(\overline{P}_\tau) = \mathcal{W}_\tau - \sum_{k=1}^{n} \widetilde{c}_{\tau k} q_k - \sum_{k=1}^{n} \overline{c}_{\tau k} q_k + r_q, \\ Tr(P_\tau) = Tr(\Pi_\tau \Psi'' \Sigma''_\tau \Pi_\tau^T). \end{aligned}$$

Considering Π_τ is a random permutation matrix, we have

$$Tr(P_\tau) = Tr(\Pi_\tau \Psi'' \Sigma''_\tau \Pi_\tau^T) = Tr(\Psi'' \Sigma''_\tau).$$

Moreover, due to π_γ is a permutation vector and

$$\Psi''(\gamma_{row}) = \pi_\gamma(\Psi'(\gamma_{row})), \Sigma''_\tau(\gamma_{col}) = \pi_\gamma\left(\Sigma'_\tau(\gamma_{col})^T\right)^T,$$

the value of $\Psi''(\gamma_{row}) \cdot \Sigma''_\tau(\gamma_{col})$ is equal to the value of $\Psi'(\gamma_{row}) \cdot \Sigma'_\tau(\gamma_{col})$, $1 \le \gamma \le n+2$, obviously,

$$Tr(\Psi'' \Sigma''_\tau) = Tr(\Psi' \Sigma'_\tau).$$

Then the sum of matrix $\Psi' \Sigma'_\tau$'s diagonal entries can be easily computed as

$$Tr(\Psi' \Sigma'_\tau) = \mathcal{R} + \sum_{j=1}^{n} \vec{\varphi}_j \cdot \vec{\sigma}_{\tau j},$$

where $\mathcal{R}$ is a random number, $\vec{\varphi}_j = q_j \cdot (q_1, q_2, \ldots, q_n)$ and $\vec{\sigma}_{\tau j} = \Sigma_\tau^{-1}(j_{col})^T = (\sigma_{\tau j}^{(1)}, \ldots, \sigma_{\tau j}^{(n)})$. Thus,

$$Tr(\Psi' \Sigma'_\tau) = \mathcal{R} + \sum_{j=1}^{n} q_j \cdot \left(\sum_{k=1}^{n} q_k \cdot \sigma_{\tau j}^{(k)} \right).$$

As mentioned above, D_τ can be denote as

$$\begin{aligned} D_\tau =& Tr(\widetilde{P}_\tau) + Tr(\overline{P}) + Tr(P\tau) \\ =& \mathcal{W}_\tau + r_q + \mathcal{R} - \sum_{k=1}^{n} \widetilde{c}_{\tau k} q_k - \sum_{k=1}^{n} \overline{c}_{\tau k} q_k \\ &+ \sum_{j=1}^{n} q_j \cdot \left(\sum_{k=1}^{n} q_k \cdot \sigma_{\tau j}^{(k)} \right) \\ =& \vec{c}_\tau \cdot \Sigma_\tau^{-1} \cdot \vec{c}_\tau^T - \vec{q} \cdot \Sigma_\tau^{-1} \cdot \vec{c}_\tau^T - \vec{c}_\tau \cdot \Sigma_\tau^{-1} \cdot \vec{q}^T \\ &+ \vec{q} \cdot \Sigma_\tau^{-1} \cdot \vec{q}^T + r_q + \mathcal{R} \\ =& D_{ma}^2(\vec{q}_\omega, \vec{c}_\tau) + r_q + \mathcal{R}. \end{aligned}$$

Finally, it is quite clear that $D_{\tau_1} - D_{\tau_2} = D_{ma}^2(\vec{q}_\omega, \vec{c}_{\tau_1}) - D_{ma}^2(\vec{q}_\omega, \vec{c}_{\tau_2})$, meanwhile, considering that $D_{ma}(\vec{q}_\omega, \vec{c}_\tau) \geq 0$, the correctness of *2)* is proved, and it can further prove the correctness of TAMMIE. □

7 Security Analysis

In this section, we prove that our basic scheme PMDC can resist KPA, and analyze the security of TAMMIE to check whether it can satisfy the privacy requirements described in Section 3.3.

7.1 Security analysis of PMDC

In this subsection, we firstly define a leakage function $\mathcal{L}$, and then introduce the real and ideal environments respectively for subsequent formal security definition and analysis.

Note. In order to provide similar medical image retrieval, PMDC leaks the "closeness" and final order of encrypted queries (i.e., ***Size Pattern***, ***Access Pattern*** and ***Search Pattern***). Thus, an optimal security notion for PMDC would be a natural relaxation of the standard IND-CPA security definition prohibiting queries that trivially exploit this leakage of closeness and order. The optimal security is called *indistinguishability under closeness-order-pattern chosen-plaintext attack* (IND-CLO-CPA) similar to [39], [40]. However, the adversary can read more information from CLO, which makes PMDC cannot achieve the standard IND-CPA security. Here, we prove it is IND-KPA secure.

7.1.1 Leakage function

PMDC is essentially equivalent to a searchable encryption scheme, as the basic part of security analysis in searchable encryption, leakage function, which can describe all possible information leaked during the whole querying process, should be defined at first. Informally, based on the encrypted samples $\{E(\vec{s}_1), E(\vec{s}_2), \ldots, E(\vec{s}_N)\}$ and the encrypted queries $E(\vec{q}_\omega), 1 \leq \omega \leq \Omega$, the leakage function $\mathcal{L}$ (***default*** information leakage) can be summarized as following aspects:

- ***Size Pattern:*** The cloud server knows the total number of encrypted samples in the database (N), the total number of encrypted queries submitted by users (Ω), and the dimensions of encrypted samples/queries $(n+2) \times (n+2)$.
- ***Access Pattern:*** The cloud server reveals each sort result $\{D_{\omega 1'}, D_{\omega 2'}, \ldots, D_{\omega N'}\}$ returned for the query q_ω, assuming there exists a mapping $\nu_\omega : i' \to i$, $i \in [1, N]$, $D_{\omega i'}$ represents the size of the Mahalanobis distance between $\vec{q}_\omega$ and $\vec{s}_{\nu_\omega(i')}$. Besides, the difference $F_{i'} = D_{ma}(\vec{q}_\omega, \vec{s}_{\nu_\omega((i+1)')}) - D_{ma}(\vec{q}_\omega, \vec{s}_{\nu_\omega(i')})$ is able to be calculated by $D_{\omega(i+1)'} - D_{\omega i'}$, $F_{i'} \geq 0$.
- ***Search Pattern:*** The cloud server can learn whether an encrypted sample or cluster center is queried by two different encrypted queries.

7.1.2 Real and ideal environment

Based on $\mathcal{L}$, we prove the security of PMDC in the real and ideal experiments, and their definitions are given in detail as follows.

Real environment. The real environment for PMDC involves a stateful probabilistic polynomial time (PPT) adversary $\mathcal{A}$ and a challenger $\mathcal{C}$, and the two participants (i.e., $\mathcal{A}$ and $\mathcal{C}$) interact as follows.

- **Initialization phase:** $\mathcal{A}$ firstly constructs a database $\mathcal{D}_\mathcal{A}$ consists of p_1 n-dimensional samples in random (i.e., $\mathcal{D}_\mathcal{A} = \{\vec{s}_i\}_{i=1}^{p_1}$), then calculates the $\mathcal{D}_\mathcal{A}$'s covariance matrix $\Sigma_\mathcal{A}$ and sends it with $\mathcal{D}_\mathcal{A}$ to $\mathcal{C}$.
- **Setup phase:** $\mathcal{C}$ runs **KeyGen**(n) to create a secret key $sk = \{M_1, M_2\}$ and keeps it in private. Then, $\mathcal{C}$ runs **DataEnc**$(\mathcal{D}_\mathcal{A}, \Sigma_\mathcal{A}, sk)$ to encrypt all samples in $\mathcal{D}_\mathcal{A}$ into $\{E(\vec{s}_i)\}_{i=1}^{p_1}$, where $E(\vec{s}_i) = \{E(\widetilde{\boldsymbol{s}}_i), E(\overline{\boldsymbol{s}}_i)\}$.
- **Query phase 1:** $\mathcal{A}$ adaptively chooses a number of queries $\{\vec{q}_\omega\}_{\omega=1}^{p_2}$ and sends them to $\mathcal{C}$. In response, for $1 \leq \omega \leq p_2$, $\mathcal{C}$ runs **TrapGen**$(\vec{q}_\omega, sk)$ to encrypt each query $\vec{q}_\omega$ into $E(\vec{q}_\omega) = \{E(\widetilde{\boldsymbol{q}}_\omega), E(\overline{\boldsymbol{q}}_\omega)\}$, and returns $E(\vec{q}_\omega)_{\omega=1}^{p_2}$ to $\mathcal{A}$.
- **Challenge phase:** $\mathcal{C}$ returns the encrypted database consists of $\{E(\vec{s}_i)\}_{i=1}^{p_1}$ to $\mathcal{A}$.
- **Query phase 2:** In this phase, $\mathcal{A}$ can also adaptively choose a number $(p_3 - p_2)$ of queries $\{\vec{q}_\omega\}_{\omega=p_2+1}^{p_3}$ and submit them to $\mathcal{C}$. Then, same as **Query phase 1**, $\mathcal{A}$ can receive $\{E(\vec{q}_\omega)\}_{\omega=p_2+1}^{p_3}$ from $\mathcal{C}$.

Let $r_\mathcal{A}$ denote the internal random bits used by $\mathcal{A}$ in the real environment, and $\text{View}_\mathcal{A}^{\text{Real}}$ denote the ensemble $(\{E(\vec{s}_i)\}_{i=1}^{p_1}, \{E(\vec{q}_\omega)\}_{\omega=1}^{p_3}, r_\mathcal{A})$. $\text{View}_\mathcal{A}^{\text{Real}}$ is essentially the view of $\mathcal{A}$ in the above-described real environment.

Ideal environment. The ideal environment for PMDC involves a stateful PPT adversary $\mathcal{A}$ and a simulator $\mathcal{S}$ with leakage function $\mathcal{L}$, and the two participants (i.e., $\mathcal{A}$ and $\mathcal{S}$) interact as follows.

- **Initialization phase:** $\mathcal{A}$ firstly constructs a database $\mathcal{D}_\mathcal{A}$ consists of p_1 n-dimensional samples in random (i.e., $\mathcal{D}_\mathcal{A} = \{\vec{s}_i\}_{i=1}^{p_1}$), then calculates the $\mathcal{D}_\mathcal{A}$'s covariance matrix $\Sigma_\mathcal{A}$ and sends it with $\mathcal{D}_\mathcal{A}$ to $\mathcal{S}$.
- **Setup phase:** For $1 \leq i \leq p_1$, $\mathcal{S}$ randomly generates two $(n+2) \times (n+2)$ matrices $\{E'(\widetilde{\boldsymbol{s}}_i), E'(\overline{\boldsymbol{s}}_i)\}$ as the ciphertext $E'(\vec{s}_i)$ of $\vec{s}_i$.
- **Query phase 1:** $\mathcal{A}$ adaptively chooses a number of queries $\{\vec{q}_\omega\}_{\omega=1}^{p_2}$ and sends them to $\mathcal{S}$. In response, based on the leakage function $\mathcal{L}$ and encrypted samples $\{E'(\vec{s}_i)\}_{i=1}^{p_1}$, $\mathcal{S}$ will generate the ciphertexts $\{E'(\vec{q}_\omega)\}_{\omega=1}^{p_2}$ for $\vec{q}_\omega, 1 \leq \omega \leq p_2$. Each randomly generated ciphertext $E'(\vec{q}_\omega) = \{E'(\widetilde{\boldsymbol{q}}_\omega), E'(\overline{\boldsymbol{q}}_\omega)\}$ should satisfy the following condition.
  Condition. $P'_{\omega i} = E'(\widetilde{\boldsymbol{s}}_i) \cdot E'(\widetilde{\boldsymbol{q}}_\omega) + E'(\overline{\boldsymbol{s}}_i) \cdot E'(\overline{\boldsymbol{q}}_\omega)$, $D'_{\omega i} = Tr(P'_{\omega i})$, thus $D'_{\omega(i+1)'} - D'_{\omega i'} = F_{i'}$, where $D'_{\omega i'} = D'_{\omega \nu_\omega(i')}$.

- **Challenge phase:** $\mathcal{S}$ sends all encrypted samples $\{E'(\vec{s}_i)\}_{i=1}^{p_1}$ to $\mathcal{A}$.
- **Query phase 2:** In this phase, $\mathcal{A}$ can also adaptively choose a number (p_3-p_2) of queries $\{\vec{q}_\omega\}_{\omega=p_2+1}^{p_3}$ and submit them to $\mathcal{S}$. Then, same as **Query phase 1**, $\mathcal{S}$ will send $\{E'(\vec{q}_\omega)\}_{\omega=p_2+1}^{p_3}$ to $\mathcal{A}$.

Again, let $r_\mathcal{A}$ denote the internal random bits used by $\mathcal{A}$ in the ideal environment, $\text{View}_{\mathcal{A},\mathcal{S}}^{\text{Ideal}}$ denote the ensemble $(\{E'(\vec{s}_i)\}_{i=1}^{p_1}, \{E'(\vec{q}_\omega)\}_{\omega=1}^{p_3}, r_\mathcal{A})$. $\text{View}_{\mathcal{A},\mathcal{S}}^{\text{Ideal}}$ is essentially the view of $\mathcal{A}$ in the above-described ideal environment.

7.1.3 Formal security definition and analysis

Based on the views of $\mathcal{A}$ in the real and ideal environments, we first give the definition of security, and then prove it.

Definition 2. *PMDC is said to be indistinguishability under the known-plaintext attack model with leakage function $\mathcal{L}$ **iff** for any PPT adversary $\mathcal{A}$, who issues a polynomial number of encrypted database samples and encrypted queries, there exists an efficient simulator $\mathcal{S}$ such the advantage of $\mathcal{A}$ in distinguishing the views of real and ideal environments is negligible, i.e., the function $Adv_\mathcal{A}^{PMDC}(n) = |Pr[View_\mathcal{A}^{Real} = 1] - Pr[View_{\mathcal{A},\mathcal{S}}^{Ideal} = 1]|$ is a negligible function in the security parameter n.*

Theorem 1. *PMDC is indistinguishability under known-plaintext attack model with $\mathcal{L}$.*

Proof. The security of PMDC can be proved if it can be proved that $\mathcal{A}$ has no ability to distinguish the views $\text{View}_\mathcal{A}^{\text{Real}} = \{\{E(\vec{s}_i)\}_{i=1}^{p_1}, \{E(\vec{q}_\omega)\}_{\omega=1}^{p_3}\}$ in the real environment and $\text{View}_{\mathcal{A},\mathcal{S}}^{\text{Ideal}} = \{\{E'(\vec{s}_i)\}_{i=1}^{p_1}, \{E'(\vec{q}_\omega)\}_{\omega=1}^{p_3}\}$ in the ideal environment. Due to $\{E'(\vec{s}_i)\}_{i=1}^{p_1}$ and $\{E'(\vec{q}_\omega)\}_{\omega=1}^{p_3}$ are all random $(n+2)\times(n+2)$ matrices generated by $\mathcal{S}$, distinguishing views $\text{View}_\mathcal{A}^{\text{Real}}$ and $\text{View}_{\mathcal{A},\mathcal{S}}^{\text{Ideal}}$ is equivalent to distinguish $\text{View}_\mathcal{A}^{\text{Real}}$ from random numbers. Meanwhile, the indistinguishability between the intermediate results $P_{\omega i}$ and $P'_{\omega i}$ also should be taken into consideration. Therefore, we will prove the indistinguishability of $\text{View}_\mathcal{A}^{\text{Real}}$ and $\text{View}_{\mathcal{A},\mathcal{S}}^{\text{Ideal}}$ from three cases.

Case 1. The encrypted samples $\{E(\vec{s}_i)\}_{i=1}^{p_1}$ are indistinguishable from random ciphertexts.

In PMDC, $E(\vec{s}_i)$ consists of $E(\widetilde{\boldsymbol{s}}_i) = M_1\widetilde{Q}_i\widetilde{S}_iM_2$ and $E(\overline{\boldsymbol{s}}_i) = M_1\overline{Q}_i\overline{S}_iM_2$, where M_1, M_2 are two random invertible matrices, and $\widetilde{Q}_i, \overline{Q}_i$ are two lower triangular matrices with diagonal entries set to 1 randomly generated for each $\vec{s}_i$. Besides, the elements in diagonal matrices $\widetilde{S}_i, \overline{S}_i$ also include a random number α_i. According to the derivation in [17], we can know that M_1 and M_2 cannot be figured out based on existing conditions. Without knowing these random matrices and number, $\{E(\vec{s}_i)\}_{i=1}^{p_1}$ are indistinguishable from random ciphertexts for $\mathcal{A}$.

Case 2. The encrypted queries $\{E(\vec{q}_\omega)\}_{\omega=1}^{p_3}$ are indistinguishable from random ciphertexts.

Same as **Case 1**, each $E(\vec{q}_\omega)$ contains two parts: $E(\widetilde{\boldsymbol{q}}_\omega) = M_2^{-1}\widetilde{\mathcal{Q}}_\omega\widetilde{Q}_\omega M_1^{-1}$ and $E(\overline{\boldsymbol{q}}_\omega) = M_2^{-1}\overline{\mathcal{Q}}_\omega\overline{Q}_\omega M_1^{-1}$. Both of them are made up of the inverse of M_1, M_2, a randomly-generated lower triangular matrix whose diagonal entries are 1 ($\widetilde{Q}_\omega$,$\overline{Q}_\omega$), and a diagonal matrix ($\widetilde{\mathcal{Q}}_\omega$,$\overline{\mathcal{Q}}_\omega$). Besides, $\widetilde{\mathcal{Q}}_\omega$ and $\overline{\mathcal{Q}}_\omega$ also contain two random entries $\beta, r_{\omega q}$. Therefore, restoring M_1, M_2 is also impossible in this case, for the attacker $\mathcal{A}$, $\{E(\vec{q}_\omega)\}_{\omega=1}^{p_3}$ and random ciphertexts are indistinguishable.

Case 3. The intermediate results $\{P_{\omega i}|1\le\omega\le p_3, 1\le i\le p_1\}$ are indistinguishable from random matrices with $\mathcal{L}$.

In PMDC, $P_{\omega i} = E(\widetilde{\boldsymbol{s}}_i)\cdot E(\widetilde{\boldsymbol{q}}_\omega) + E(\overline{\boldsymbol{s}}_i)\cdot E(\overline{\boldsymbol{q}}_\omega)$, that is, $P_{\omega i} = M_1(\widetilde{Q}_i\widetilde{S}_i\widetilde{\mathcal{Q}}_\omega\widetilde{Q}_\omega + \overline{Q}_i\overline{S}_i\overline{\mathcal{Q}}_\omega\overline{Q}_\omega)M_1^{-1}$. While $\vec{q}_\omega$ is chosen by $\mathcal{A}$, β, $r_{\omega q}$, $\widetilde{Q}_i$, $\overline{Q}_i$, $\widetilde{S}_i$ and $\overline{S}_i$ are still unknown such that $\mathcal{A}$ has no way to figure out M_1. In addition, due to $\widetilde{Q}_i\widetilde{S}_i\widetilde{\mathcal{Q}}_\omega\widetilde{Q}_\omega + \overline{Q}_i\overline{S}_i\overline{\mathcal{Q}}_\omega\overline{Q}_\omega$ are not a diagonal matrix any more, many random numbers are as entries in this matrix, neither $\widetilde{S}_i$ nor $\overline{S}_i$ can be recovered by $\mathcal{A}$. The detailed derivation process can refer to the Appendix B in [17], we do not describe here for page limit. Thus, except for the information leaked by $\mathcal{L}$, the intermediate results $\{P_{\omega i}|1\le\omega\le p_3, 1\le i\le p_1\}$ are indistinguishable from random matrices for $\mathcal{A}$.

In conclusion, the advantage of $\mathcal{A}$ in distinguishing $\text{View}_\mathcal{A}^{\text{Real}}$ and $\text{View}_{\mathcal{A},\mathcal{S}}^{\text{Ideal}}$ is negligible. Therefore, PMDC can be proven to be indistinguishability under known-plaintext attack model with $\mathcal{L}$. □

7.2 Security analysis of TAMMIE

In TAMMIE, we use a symmetric-key algorithm (e.g. AES) to encrypt the original images before outsourcing them to CS, and AES cannot be attacked successfully without secret key k_I has been proved in [41]. Therefore, in this subsection, we mainly focus on the privacy of indexes (i.e., feature vectors and cluster centers) and queries.

The indexes are privacy-preserving. In our threat model, CS is considered as a *honest-but-curious* third-party, thus it maybe try to analyze the encrypted feature vectors $E(\vec{s}_{\tau\kappa})$ in each cluster $\mathcal{C}_\tau$ and encrypted cluster centers EC_τ to read original information, where $1\le\tau\le c$, $1\le\kappa\le|\mathcal{C}_\tau|$. Due to $E(\vec{s}_{\tau\kappa})$ is generated by the algorithm PMDC_**DataEnc**, the security of PMDC can prevent CS from obtaining arbitrary feature vector $\vec{s}_{\tau\kappa}$ from ciphertext. Meanwhile, the cluster center can reveal the attributes of the cluster to which it belongs, thus any EC_τ should not be able to provide valid information to CS. In TAMMIE, EC_τ consists of two parts $E(\vec{c}_\tau)$ and $E(\Sigma_\tau)$, considering that $E(\vec{c}_\tau)$ is the result of $\vec{c}_\tau$ being encrypted by PMDC_**DataEnc**, here we focus on analyzing whether CS can recover Σ_τ from $E(\Sigma_\tau)$, where $E(\Sigma_\tau) = M_3\Sigma''_\tau\Pi_\tau^T M_4$. Specifically, M_3 and M_4 are two random matrices unknown to CS, Π_τ is a random permutation matrix owned by IO and QUs, and each column in Σ''_τ is a random permutation of the corresponding column in Σ'_τ (Σ'_τ is generated by Σ_τ being added $(4n+4)$ random numbers). Obviously, without knowing M_3, M_4 and permutation matrices/vectors, CS has no way to achieve any cluster's covariance matrix Σ_τ. Therefore, we think the indexes are privacy-preserving.

The queries are privacy-preserving. When receiving a trapdoor TD from QU, CS will try to analyze it to read the sensitive data of QU, thus TD should guarantee that $\vec{q}$ is kept secret from CS. Like EC_τ, TD also consists of two parts, that is, $E(\vec{q})$ and $\{E_\tau(\Psi)|_{\tau=1}^{c}\}$. $E(\vec{q})$ is calculated by running PMDC_**TrapGen**, thus its security depends on the security of PMDC. For $1\le\tau\le c$, $E_\tau(\Psi) = M_4^{-1}\Pi_\tau\Psi''M_3^{-1}$, without knowing M_3, M_4, Π_τ as well as the random matrices

and permutation vectors used in the process of converting $\vec{q}$ to Ψ'', it is impossible for CS to dig out any meaningful information from $\{E_\tau(\Psi)|_{\tau=1}^{c}\}$.

8 Performance Evaluation

In this section, we analyze the performance of TAMMIE in terms of theoretical analysis and experimental evaluations, and make comparisons with the state-of-the-art similar work [9], [16]. For the theoretical analysis, we mainly focus on the communication and computation overheads. After that, we perform empirical experiments on the real-world and synthetic datasets to demonstrate the accuracy and efficiency of our TAMMIE.

TABLE 2
Theoretical performance of various schemes

Schemes	Computation complexity		
	IndexEnc	**TrapGen**	**Search**
VFIRM [9]	$\mathcal{O}(N(n+d)^2)$	$\mathcal{O}((n+d)^2)$	$\mathcal{O}(N(n+d))$
SEI [16]	$\mathcal{O}(4N_s n^2)$	$\mathcal{O}(4n^2)$	$\mathcal{O}(2C_s n)$
TAMMIE	$\mathcal{O}(N_t(n+2)^3)$	$\mathcal{O}(c'(n+2)^3)$	$\mathcal{O}(C_t(n+2)^2)$
Schemes	**Communication complexity**		
	IndexEnc	**TrapGen**	**Search**
VFIRM [9]	$\mathcal{O}(N(n+d))$	$\mathcal{O}(n+d)$	$\mathcal{O}(k)$
SEI [16]	$\mathcal{O}(2N_s n)$	$\mathcal{O}(2n)$	$\mathcal{O}(k)$
TAMMIE	$\mathcal{O}(N_t(n+2)^2)$	$\mathcal{O}(c'(n+2)^2)$	$\mathcal{O}(k)$

Notations. N is the number of medical images; n is the dimension of indexes/queries; d is the extended dimension; N_s/N_t is the number of indexes needed to be encrypted in SEI/TAMMIE; C_s/C_t is the number of encrypted indexes needed to be calculated in SEI/TAMMIE; $c' = c + 1$, where c is the number of cluster centers; k is the number of returned similar images.

8.1 Theoretical analysis

In TABLE 2, we compare the complexity of our proposed scheme with VFIRM [9] and SEI [16], which are two state-of-the-art encrypted images retrieval schemes similar to ours. The comparison contains three core phases: **IndexEnc**, in which DO encrypts and outsources the indexes; **TrapGen**, in which QUs encrypt and send their queries; and **Search**, in which CS finds and returns the top-k similar images. We assume that there is only one non-revoked IO and QU, and our analysis is mainly conducted from the two perspectives of computation and communication. Some notations used in theoretical analysis have been described in TABLE 1, and the definitions of others are as follows: N_s represents the number of images and tree nodes in SEI; N_t represents the number of images and clusters in TAMMIE ($N_t = N + c$); C_s/C_t represents the number of encrypted indexes (including tree nodes or cluster centers) needed to be computed in SEI (TAMMIE); k represents the number of returned similar medical images.

Computation. For expression simplicity, the computation complexity taken to compute the inner product of two n-dimensional vectors is $\mathcal{O}(n)$; thereby, the multiplication of an n-dimensional vector and an $n \times n$-dimensional matrix has the computation complexity of $\mathcal{O}(n^2)$, and the $n \times n$-dimensional matrix multiplication has the computation complexity of $\mathcal{O}(n^3)$. In VFIRM, taking no consideration about its operations of verification, the indexes are extended to $(n+d)$-dimensional and encrypted by executing the multiplication of vectors and matrices in **IndexEnc** and **TrapGen**, thus the computation complexity of **IndexGen** and **TrapGen** can be denoted as $\mathcal{O}(N(n+d)^2)$ and $\mathcal{O}((n+d)^2)$ in respective. Before finding the top-k similar images, CS needs to calculate inner product N times, compared with inner product, the time cost of sorting is negligible, thus the computation complexity of **Search** can be regarded as $\mathcal{O}(N(n+d))$. The encryption method of SEI is similar to that of VFIRM, but SEI extends the dimension of indexes to $2n$ and constructs a hierarchical index tree to improve retrieval efficiency, which increases the computation complexity of **IndexEnc** to $\mathcal{O}(4N_s n^2)$ and reduces the computation complexity of **Search** to $\mathcal{O}(2C_s n)$. In addition, the computation complexity of **TrapGen** becomes $\mathcal{O}(4n^2)$. In TAMMIE, the dimension of indexes is expanded to $n + 2$, the encryption is achieved by matrix multiplication, and the sorting is achieved by computing the traces. It also introduces FCM-M algorithm to reduce the computation overhead of **Search**, but this operation leads to extra encryption for cluster centers in **IndexEnc** and **Query**. Therefore, the computation complexity of **IndexEen**, **TrapGen** and **Search** in TAMMIE are $\mathcal{O}(N_t(n+2)^3)$, $\mathcal{O}(c'(n+2)^3)$ and $\mathcal{O}(C_t(n+2)^2)$ respectively, where $c' = c + 1$.

Communication. For ease of description, we assume the communication complexity of an n-dimensional vector and an $n \times n$-dimensional matrix are $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$, respectively. It is obviously that the outputs of VFIRM and SEI in **IndexEnc** and **TrapGen** are vectors, and the outputs of TAMMIE in **IndexEnc** and **TrapGen** are matrices. Considering the additional nodes of a tree and centers of the clusters, SEI and TAMMIE need more communication overheads in **IndexEnc**. Therefore, in **IndexEnc**, the communication complexity of VFIRM, SEI and TAMMIE are $\mathcal{O}(N(n+d))$, $\mathcal{O}((2N_s n)$ and $\mathcal{O}((N_t(n+2)^2)$ respectively; in **TrapGen**, the communication complexity of VFIRM, SEI and TAMMIE are $\mathcal{O}(n+d)$, $\mathcal{O}(2n)$ and $\mathcal{O}(c'(n+2)^2)$ respectively. Since the purpose of these schemes is to return k encrypted images to a query user, their communication complexity of **Search** are all $\mathcal{O}(k)$.

In summary, whether it is computation or communication complexity, TAMMIE does not show an advantage compared to the two state-of-the-art schemes. However, the sacrifice of complexity improves the security of our proposed scheme. As far as we know, VFIRM and SEI can only resist COA, but TAMMIE has the ability to resist KPA.

8.2 Experimental evaluations

In order to evaluate the search accuracy and efficiency of our proposed schemes, we implement TAMMIE on a computer (Intel i5 2.0GHz four-core processor, 16GB RAM, macOS Big Sur system) with Python programming language. Moreover, all performance evaluations in this section are conducted on two real-world datasets and a randomly generated synthetic dataset. The details of the two datasets are given as follows.

- Real-world datasets. 1) **IDRiD** [42] contains 413 retinal fundus images, of which 177 images have no risk of macular edema, and the remaining 236 images

may catch macular edema; 2) **COVID** randomly selected from [43] and [44] contains 1000 lung images, 400 of which belong to COVID-19 positive cases and 600 belong to normal people.

- Synthetic dataset. Synthetic dataset is randomly generated to test the factors how to affect the performance of TAMMIE, SEI and VFIRM, which consists of 10000 indexes with different dimensions from 8 to 128.

8.2.1 Accuracy

Based on the two real-world datasets, we use Precision at top-k (P@k), P@k= $num\_positive/k$, to measure the search accuracy of TAMMIE, SEI and VFIRM, where $num\_positive$ denotes the number of positive images in the top-k results. For fair comparison, we all use CNN to extract feature vectors from medical images and use PCA/PCA-ITQ method to reduce the dimension to 8, 16, 24, 48, 96, due to the similarity metric used in VFRIM is Hamming distance, VFRIM needs utilize PCA-ITQ to compress vectors into binary. Besides, the number of clusters in TAMMIE is set to 10 and the number of bottom nodes of the tree in SEI is set to 100.

TABLE 3
P@k under different dimensions of feature vectors with $k = 5$

n	IDRiD			COVID		
	TAMMIE	SEI	VFIRM	TAMMIE	SEI	VFIRM
8	**0.797**	0.791	0.543	**0.938**	0.929	0.520
16	**0.804**	0.795	0.543	**0.944**	0.937	0.520
24	**0.800**	0.787	0.543	**0.946**	0.941	0.520
48	**0.801**	0.798	0.543	**0.952**	0.944	0.520
96	**0.800**	0.795	0.543	**0.953**	0.948	0.520

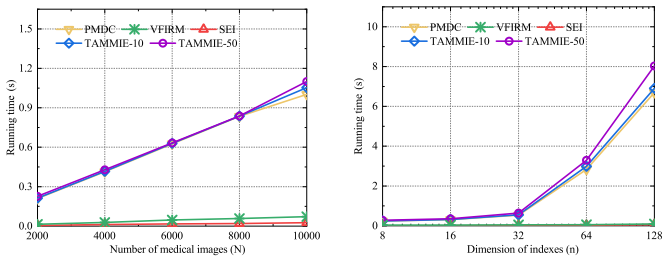
TABLE 4
P@k under different number of returned images with $n = 24$

k	IDRiD			COVID		
	TAMMIE	SEI	VFIRM	TAMMIE	SEI	VFIRM
2	**0.870**	0.869	0.500	**0.970**	**0.970**	0.500
3	**0.835**	0.827	0.524	**0.959**	0.952	0.533
5	**0.800**	0.787	0.543	**0.946**	0.941	0.520
7	**0.776**	0.773	0.551	**0.940**	0.930	0.514
9	**0.763**	0.755	0.540	**0.935**	0.927	0.511

The search accuracy mainly depends on the dimension of feature vectors (n) and the number of returned images (k). In TABLE 3, we set $k = 5$, vary n from 8 to 96, and find that the search accuracy of VFIRM is not affected, but the search accuracy of TAMMIE and SEI are affected by changes in n. In TABLE 4, we set $n = 24$, vary k from 2 to 9 to show the effect of k on the search accuracy. It can be seen that the accuracy of TAMMIE and SEI decreases as k increases, and the accuracy of VFIRM is also influenced. The test results demonstrate that VFIRM does not apply to the retrieval of medical images (accuracy around 50%), TAMMIE performs better than SEI over the two medical images datasets.

8.2.2 Efficiency

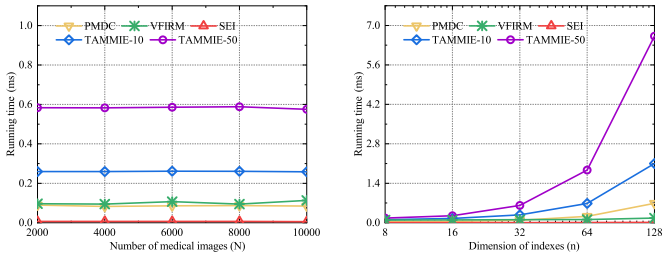
In this subsection, we use the synthetic dataset to conduct experiments and make a comparison with SEI and VFIRM from three phases, i.e., **IndexEnc**, **TrapGen** and **Query**. Specifically, in our proposed scheme, we set the number of clusters (c) to 1, 10, 50. When $c = 1$, the efficiency of TAMMIE is same as that of PMDC, thus we denote them as PMDC, TAMMIE-10 and TAMMIE-50 respectively. In SEI, the number of nodes at the bottom is set to 100 to facilitate the tree construction. In VFIRM, we adopt the same dimension expansion method (d) as the performance evaluation in [9], which just considers its access control.



(a) $n = 32$ (b) $N = 6000$

Fig. 3. Average running time of **IndexEnc** (test 100 times).

IndexEnc. As analyzed in Section 8.1, the computation cost of **IndexEnc** is mainly affected by N and n. In Fig. 3(a), we plot the average running time of **IndexEnc** varying with N ranges from 2000 to 10000, where $n = 32$. We can see that the running time of **IndexEnc** in TAMMIE (i.e., PMDC, TAMMIE-10 and TAMMIE-50) and the two comparison scheme linearly grows with the increase of N. And the increase of c has a relatively small impact on the running time. Given $N = 6000$, Fig. 3(b) is drawn with n from 8 to 128. We can observe that the running time of all schemes increases with n. When n increases, the larger c is, the longer the running time will be, this is caused by the increase encryption of cluster centers. Fig. 3 shows that both SEI and VFIRM are more efficient than our scheme, but **IndexEnc** does not require real-time processing, the running time of TAMMIE is acceptable.

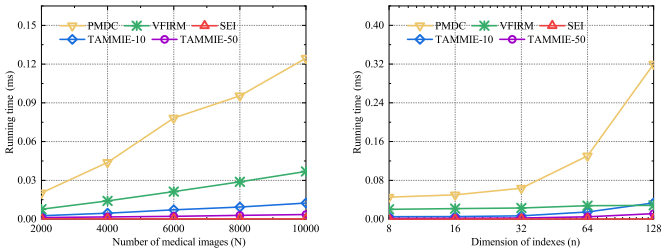


(a) $n = 32$ (b) $N = 6000$

Fig. 4. Average running time of **TrapGen** (test 100 times).

TrapGen. In Fig. 4, we plot the average running time of **TrapGen** varying with N and n, respectively. When N ranges from 2000 to 10000 ($n = 32$), Fig. 4(a) shows that the running time of **TrapGen** in TAMMIE, SEI and VFIRM is not affected, but the running time increases with the increase of c due to more cluster centers needed to be calculated. In Fig 4(b), we set $N = 6000$, vary n from 8 to 128, it can be observed that the running time of all schemes grows with the increase of n, and the clustering slows down the speed of **TrapGen**. In this phase, SEI still performs better than

TAMMIE, but when $n \leq 32$, PMDC performs better than VFIRM due to the chosen of d in VFIRM. When $n = 32$, the running time of PMDC, TAMMIE-10 and TAMMIE-50 are 0.7ms, 2.1ms and 6.6ms respectively.



(a) $n = 32$ (b) $N = 6000$

Fig. 5. Average running time of **Query** (test 100 times).

Query. The computation cost of **Query** is affected by N and d. In Fig. 5(a), we plot the average running time of **Query** varying with N ranges from 2000 to 10000, where $n = 32$. We can see that the running time of **Query** in TAMMIE and the two comparison scheme linearly grows with the increase of N. Given $N = 6000$, Fig. 5(b) is drawn with n from 8 to 128. We can observe that the running time of all schemes increases with the increase of n. Meanwhile, from Fig. 5, it can be observed that SEI still performs best, but the running time of **Query** in TAMMIE decreases with the increase of c, when $c = 10, 50$, TAMMIE is more efficient than VFIRM and close to SEI.

9 Conclusion

In this paper, we first designed a novel Mahalanobis distance secure comparison method, called PMDC. Then, based on PMDC, we proposed an accurate and privacy-preserving medical image retrieval scheme named TAMMIE by introducing FCM-M clustering algorithm. With TAMMIE, the medical institution can securely outsource its medical image dataset to a cloud server, and the physicians can send their trapdoors to request retrieval services over the outsourced data. Detailed security analysis showed that TAMMIE can achieve indistinguishability secure under known-plaintext attack, and extensive experiments were conducted to demonstrate its high accuracy and practicality. In the future work, we will take the improvement of efficiency into consideration.

Acknowledgments

This work was supported by National Key R&D Program of China (2021YFB3101300), National Natural Science Foundation of China (61972304 and 61932015), and Shaanxi Innovation Team Project (2018TD-007).

References

[1] Ş. Öztürk, "Stacked auto-encoder based tagging with deep features for content-based medical image retrieval," *Expert Systems with Applications*, vol. 161, p. 113693, 2020.

[2] Y. Zhang, Y. Wei, Q. Wu, P. Zhao, S. Niu, J. Huang, and M. Tan, "Collaborative unsupervised domain adaptation for medical image diagnosis," *IEEE Transactions on Image Processing*, vol. 29, pp. 7834–7844, 2020.

[3] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009*. ACM, 2009, pp. 139–152.

[4] Z. Xia, N. N. Xiong, A. V. Vasilakos, and X. Sun, "EPCBIR: an efficient and privacy-preserving content-based image retrieval scheme in cloud computing," *Information Sciences*, vol. 387, pp. 195–204, 2017.

[5] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.

[6] J. Yuan, S. Yu, and L. Guo, "SEISA: secure and efficient encrypted image search with access control," in *Proceedings of the IEEE Conference on Computer Communications, INFOCOM 2015*. IEEE, 2015, pp. 2083–2091.

[7] X. Li, Q. Xue, and M. C. Chuah, "CASHEIRS: cloud assisted scalable hierarchical encrypted based image retrieval system," in *Proceedings of the IEEE Conference on Computer Communications, INFOCOM 2017*. IEEE, 2017, pp. 1–9.

[8] Y. Li, J. Ma, Y. Miao, Y. Wang, T. Yang, X. Liu, and K.-K. R. Choo, "Traceable and controllable encrypted cloud image search in multi-user settings," *IEEE Transactions on Cloud Computing*, 2020.

[9] Q. Tong, Y. Miao, L. Chen, J. Weng, X. Liu, K.-K. R. Choo, and R. Deng, "Vfirm: Verifiable fine-grained encrypted image retrieval in multi-owner multi-user settings," *IEEE Transactions on Services Computing*, 2021.

[10] R. Li, A. X. Liu, Y. Liu, H. Xu, and H. Yuan, "Insecurity and hardness of nearest neighbor queries over encrypted data," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 1614–1617.

[11] J. Yuan and Y. Tian, "Practical privacy-preserving mapreduce based k-means clustering over large-scale dataset," *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 568–579, 2019.

[12] X. Wang, J. Ma, X. Liu, and Y. Miao, "Search in my way: Practical outsourced image retrieval framework supporting unshared key," in *Proceedings of the IEEE Conference on Computer Communications, INFOCOM 2019*. IEEE, 2019, pp. 2485–2493.

[13] Y. Li, J. Ma, Y. Miao, L. Liu, X. Liu, and K.-K. R. Choo, "Secure and verifiable multikey image search in cloud-assisted edge computing," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5348–5359, 2021.

[14] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[15] Y. Zhu, J. Yu, and C. Jia, "Initializing k-means clustering using affinity propagation," in *2009 Ninth International Conference on Hybrid Intelligent Systems*, vol. 1. IEEE, 2009, pp. 338–343.

[16] Y. Li, J. Ma, Y. Miao, Y. Wang, X. Liu, and K.-K. R. Choo, "Similarity search for encrypted images in secure cloud computing," *IEEE Transactions on Cloud Computing*, 2020.

[17] Q. Wang, S. Hu, K. Ren, M. He, M. Du, and Z. Wang, "Cloudbi: Practical privacy-preserving outsourcing of biometric identification in the cloud," in *Proceedings of the 20th European Symposium on Research in Computer Security, ESORICS 2015*, ser. Lecture Notes in Computer Science, G. Pernul, P. Y. A. Ryan, and E. R. Weippl, Eds., vol. 9327. Springer, 2015, pp. 186–205.

[18] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, "Enabling search over encrypted multimedia databases," in *Proceedings of the Media Forensics and Security, SPIE 2009*, E. J. Delp, J. Dittmann, N. D. Memon, and P. W. Wong, Eds., vol. 7254. SPIE, 2009, p. 725418.

[19] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu, "Secure image retrieval through feature protection," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009*. IEEE, 2009, pp. 1533–1536.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

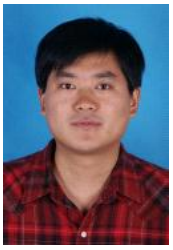
[21] C. Hsu, C. Lu, and S. Pei, "Image feature extraction in encrypted domain with privacy-preserving SIFT," *IEEE Transactions on Image Process*, vol. 21, no. 11, pp. 4593–4607, 2012.

[22] L. Zhang, T. Jung, K. Liu, X.-Y. Li, X. Ding, J. Gu, and Y. Liu, "Pic: Enable large-scale privacy preserving content-based image search on cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 11, pp. 3258–3271, 2017.

[23] C. Guo, S. Su, K.-K. R. Choo, and X. Tang, "A fast nearest neighbor search scheme over outsourced encrypted medical images," *IEEE*

Transactions on Industrial Informatics, vol. 17, no. 1, pp. 514–523, 2018.
[24] Y. Elmehdwi, B. K. Samanthula, and W. Jiang, "Secure k-nearest neighbor query over encrypted data in outsourced environments," in *2014 IEEE 30th International Conference on Data Engineering*. IEEE, 2014, pp. 664–675.
[25] W. Lu, A. L. Varna, and M. Wu, "Confidentiality-preserving image search: A comparative study between homomorphic encryption and distance-preserving randomization," *IEEE Access*, vol. 2, pp. 125–141, 2014.
[26] M. Shen, Y. Deng, L. Zhu, X. Du, and N. Guizani, "Privacy-preserving image retrieval for medical iot systems: A blockchain-based approach," *IEEE Network*, vol. 33, no. 5, pp. 27–33, 2019.
[27] A. F. S. Devaraj, G. Murugaboopathi, M. Elhoseny, K. Shankar, K. Min, H. Moon, and G. P. Joshi, "An efficient framework for secure image archival and retrieval system using multiple secret share creation scheme," *IEEE Access*, vol. 8, pp. 144 310–144 320, 2020.
[28] P. Vepakomma, J. Balla, and R. Raskar, "Privatemail: Supervised manifold learning of deep features with privacy for image retrieval," *arXiv preprint arXiv:2102.10802*, 2021.
[29] Y. Liu, Z. Ma, X. Liu, S. Ma, and K. Ren, "Privacy-preserving object detection for medical images with faster r-cnn," *IEEE Transactions on Information Forensics and Security*, 2019.
[30] C. Zhang, L. Zhu, S. Zhang, and W. Yu, "TDHPPIR: an efficient deep hashing based privacy-preserving image retrieval method," *Neurocomputing*, vol. 406, pp. 386–398, 2020.
[31] H. Lu, M. Zhang, X. Xu, Y. Li, and H. T. Shen, "Deep fuzzy hashing network for efficient image retrieval," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 1, pp. 166–176, 2021.
[32] O. Goldreich, *Foundations of cryptography: volume 2, basic Applications*. Cambridge university press, 2009.
[33] H. Delfs and H. Knebl, *Introduction to Cryptography - Principles and Applications, Third Edition*, ser. Information Security and Cryptography. Springer, 2015.
[34] W. Xiangyu, J. Ma, M. Yinbin, X. Liu, and Y. Ruikang, "Privacy-preserving diverse keyword search and online pre-diagnosis in cloud computing," *IEEE Transactions on Services Computing*, 2019.
[35] G. J. McLachlan, "Mahalanobis distance," *Resonance*, vol. 4, no. 6, pp. 20–26, 1999.
[36] N. A. H. Haldar, F. A. Khan, A. Ali, and H. Abbas, "Arrhythmia classification using mahalanobis distance based improved fuzzy c-means clustering for mobile health monitoring systems," *Neurocomputing*, vol. 220, pp. 221–235, 2017.
[37] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, p. 417, 1933.
[38] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 98–113, 1997.
[39] P. Grubbs, K. Sekniqi, V. Bindschaedler, M. Naveed, and T. Ristenpart, "Leakage-abuse attacks against order-revealing encryption," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 655–672.
[40] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, "Enabling efficient and geometric range query with access control over encrypted spatial data," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 870–885, 2018.
[41] P. Hämäläinen, T. Alho, M. Hännikäinen, and T. D. Hämäläinen, "Design and implementation of low-area and low-power AES encryption hardware core," in *Proceedings of the 9th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, DSD 2006*. IEEE Computer Society, 2006, pp. 577–583.
[42] P. Porwal, S. Pachade, R. Kamble, M. Kokare, G. Deshmukh, V. Sahasrabuddhe, and F. Meriaudeau, "Indian diabetic retinopathy image dataset (IDRiD)," https://dx.doi.org/10.21227/H25W98.
[43] M. E. H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. R. Islam, M. S. Khan, A. Iqbal, N. A. Emadi, M. B. I. Reaz, and M. T. Islam, "Can ai help in screening viral and covid-19 pneumonia?" *IEEE Access*, vol. 8, pp. 132 665–132 676, 2020.
[44] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughaier, M. S. Khan *et al.*, "Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images," *Computers in biology and medicine*, vol. 132, p. 104319, 2021.

Dan Zhu currently is a Ph.D candidate in Xidian University. She received the B.S. degree with the School of Telecommunications Engineering from Xidian University, xi'an, China, in 2017. Her research interests include applied cryptography, data security and privacy.



Hui Zhu (M'13-SM'19) received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2003 and 2009, respectively, and the M.S. degree from Wuhan University, Wuhan, China, in 2005. In 2013, he was with School of Electrical and Electronics Engineering, Nanyang Technological University as a Research Fellow. Since 2016, he has been the professor in the School of Cyber Engineering, Xidian University, China. His research interests include the areas of applied cryptography, data security and privacy.

Xiangyu Wang currently is a Ph.D candidate in Xidian University. He received the B.S. degree with the School of Cyber Engineering from Xidian University, xi'an, China, in 2017. His research interests include data security and secure computation outsourcing.

Rongxing Lu (S'09-M'11-SM'15-F'21) is an associate professor at the Faculty of Computer Science, University of New Brunswick, Canada. Before that, he worked as an assistant professor at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore from April 2013 to August 2016. Rongxing Lu worked as a Postdoctoral Fellow at the University of Waterloo from May 2012 to April 2013. He received his PhD degree from the Department of Electrical & Computer Engineering, University of Waterloo, Canada, in 2012. Dr. Lu is an IEEE Fellow. Currently, Dr. Lu serves as the Vice-Chair (Conferences) of IEEE ComSoc CIS-TC (Communications and Information Security Technical Committee). His research interests include applied cryptography, privacy enhancing technologies, and IoT-Big Data security and privacy.

Dengguo Feng received the B.S. degree from Shaanxi Normal University, Xi'an, China, in 1988, the M.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 1993 and 1995, respectively. He is currently a Professor with the Institute of Software, Chinese Academy of Sciences, Beijing, China. He is a recipient of China National Funds for Distinguished Young Scientists. He is the Vice-Chairmen of Chinese Association for Cryptologic Research and a Steering Committee Member of Information Technology in National High-Tech R&D Program of China.