Efficient and Accurate Cloud-Assisted Medical Pre-Diagnosis With Privacy Preservation

Dan Zhu, Hui Zhu, *Senior Member, IEEE*, Cheng Huang, *Member, IEEE*, Rongxing Lu, *Fellow, IEEE*, Dengguo Feng, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—The emergence of cloud computing enables various healthcare institutions to outsource pre-diagnostic models and provide timely and convenient services for patients. However, healthcare institutions and patients have serious concerns about potential privacy leakage as cloud servers cannot be fully trusted. In this paper, a privacy-preserving cloud-assisted medical pre-diagnosis scheme, named NAIAD, is proposed, where patients can securely query the outsourced model and obtain their pre-diagnostic results. Specifically, the pre-diagnostic model is constructed on k-Nearest Neighbor (kNN), and Mahalanobis Distance (MD) is chosen as the similarity metric to achieve high accuracy. Accordingly, a secure MD-based comparison method (SMDC) is designed based on a matrix encryption technique. The method is a basic module of NAIAD that enables cloud servers to compare encrypted medical records and achieve privacy-preserving kNN-based pre-diagnosis with linear complexity. To further improve the computational efficiency, medical records are first clustered and encrypted to construct a hierarchical index tree, then patients can query the tree to speed up the query process. Detailed security analysis indicates NAIAD can resist closeness-same-pattern chosen-plaintext attack, and extensive experiments on real-world and synthetic databases demonstrate NAIAD has high query efficiency and pre-diagnosis accuracy.**

***Index Terms*—Hierarchical index tree, mahalanobis distance, medical pre-diagnosis, privacy-preserving, similarity comparison.**

I. Introduction

With the popularization of cloud computing, various cloud-assisted medical services have been established and found their way into our daily life, and medical pre-diagnosis is one of the most notable services [1], [2], [3], [4]. Healthcare institutions (e.g., hospitals) are able to train appropriate pre-diagnostic models using large amount of collected medical data and outsource the models to a cloud server. Patients can send queries to the cloud server anytime and anywhere based on their personal health information (PHI), such as physiological data, and obtain the pre-diagnostic results. However, recent medical data breach events have raised serious concerns about the reliability of cloud servers, as they are maintained by profit-driven companies and are vulnerable to external attacks [5]. Considering the queries and pre-diagnostic results are sensitive PHI [6], and the pre-diagnostic models are private commercial assets, neither of them should be revealed to the cloud server directly. Therefore, taking into account the privacy preservation in cloud-assisted medical pre-diagnosis is of great importance.

To address the above-mentioned issue, several privacy-preserving medical pre-diagnosis schemes have been proposed based on multiple machine learning classifiers, such as Naïve Bayesian (NB) and Support Vector Machine (SVM) [7], [8], [9], [10], [11], [12], [13], [14], [15]. Among the most popular machine learning classifiers used for medical pre-diagnosis, k-Nearest Neighbor (kNN) has attracted great attention as it avoids the heavy training process but still guarantees relatively good classification accuracy [8], [11], [14], [16], [17]. Concretely, kNN allows patients to compare their physiological data with a large volume of medical records, then the top-k similar records and the corresponding labels can be located and returned as the pre-diagnosis results. For kNN-based pre-diagnosis, it is extremely important to choose an appropriate similarity metric based on the structure of medical records since the bad metrics can severely degrade the accuracy of pre-diagnostic results. As one of the most common similarity metrics, *euclidean distance* is usually preferred in existing schemes [8], [11], [16], [17]. As far as physiological data is concerned, data of different dimensions are related to each other, e.g., height can affect weight, age can affect blood pressure. However, euclidean distance as the similarity metric cannot well capture the dimensional correlation in kNN-based pre-diagnosis, and *Mahalanobis distance*(MD) is considered as an alternative choice [14], [15]. It can precisely measure dimensional correlation and achieve higher accuracy in medical pre-diagnosis. Nevertheless, calculating top-k similar records based on MD requires the introduction of a covariance matrix and it will bring complex calculations. To achieve privacy preservation, heavy cryptographic techniques such as homomorphic encryption cannot be straightforwardly applied due to the practicality issues. As a consequence, it is still challenging to design a lightweight, efficient, and privacy-preserving kNN-based

Manuscript received 4 April 2022; revised 24 March 2023; accepted 27 March 2023. Date of publication 3 April 2023; date of current version 14 March 2024. This work was supported by the National Natural Science Foundation of China under Grants U22B2030, 61972304, 62125205, and 62202364, in part by Shaanxi Provincial Key Research and Development Program under Grant 2023-ZDLGY-35, in part by Fundamental Research Funds for the Central Universities under Grant ZYTS23166, and Natural Sciences and Engineering Research Council (NSERC) of Canada. *(Corresponding author: Cheng Huang.)*

Dan Zhu and Hui Zhu 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).

Cheng Huang and Xuemin (Sherman) Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: cheng.huang@uwaterloo.ca; sshen@uwaterloo.ca).

Rongxing Lu is with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada (e-mail: rlu1@unb.ca).

Dengguo 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).

Digital Object Identifier 10.1109/TDSC.2023.3263974

pre-diagnosis scheme where MD is selected as the similarity metric.

In this paper, we propose an efficient and accurate cloud-assisted medical pre-diagnosis scheme, named NAIAD. To fulfill the privacy requirements, we design a secure MD comparison method, named SMDC, by tailoring an existing matrix encryption technique. The method supports the MD similarity comparison of encrypted medical records and patients' encrypted physiological data, thus it can be used as a basic module of NAIAD to achieve privacy-preserving pre-diagnosis. To further improve the query efficiency of NAIAD, medical records are first clustered and encrypted to construct a hierarchical index tree, and patients can query the tree based on an improved SMDC. By doing so, the query complexity of NAIAD can be significantly reduced. In summary, the main contributions of the paper are as follows:

- We design a secure Mahalanobis-distance similarity comparison method called SMDC. Through applying SMDC, we can compare Mahalanobis distance on ciphertexts with the same covariance matrix. Moreover, SMDC is a generic secure similarity comparison method that can be applied not only to medical pre-diagnosis, but also to other searchable schemes, such as fingerprint recognition.
- We propose an efficient and accurate cloud-assisted medical pre-diagnosis scheme with privacy preservation named NAIAD. Aiming at improving the query efficiency, a hierarchical index tree is constructed on the clustered medical records. For privacy preservation, an improved SMDC is used to support similarity comparison of encrypted clusters clustered by medical records with different covariance matrices.
- We prove the security of SMDC and NAIAD to demonstrate that they can achieve indistinguishability under closeness-same-pattern chosen-plaintext attack. Furthermore, by using Python programming, we implement SMDC, NAIAD and compare them with an existing scheme. Sufficient theoretical analysis and comparative experiments conducted on five real-world and one synthetic databases demonstrate that SMDC and NAIAD achieve high accuracy and exhibit good performance.

The remainder of this paper is organized as follows. In Section II, we present system and threat models, and identify the privacy requirements. We design SMDC in Section III, and propose NAIAD based on SMDC in Section IV. We analyze the security and performance of SMDC and NAIAD in Section V and Section VI, respectively. The related work is reviewed in Section VII followed by the conclusion in Section VIII.

II. Models and Privacy Requirements

In this section, we give the details of our system model and threat model, and identify the privacy requirements.

A. System Model

Our system model involves three main entities, i.e., a Healthcare Institution (HI), a Cloud Server (CS) and Patients (PTs) as shown in Fig. 1:

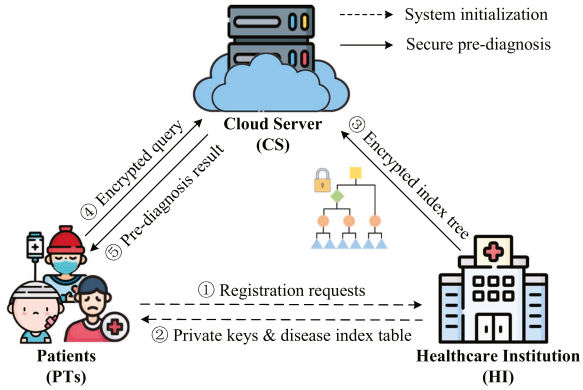


Fig. 1. System model under consideration.

- HI owns a huge dataset of medical records, and each record includes a patient's physiological data and corresponding disease. To offer efficient and privacy-preserving pre-diagnosis services, HI needs to build an encrypted hierarchical index tree over the records and outsource it to CS. In addition, HI also needs to provide PTs with registration services;
- PTs denote a collection of patients who request pre-diagnosis. Each PT $\in$ PTs must register at HI to obtain private keys and a disease index table at first, then generate the trapdoor (i.e., an encrypted query request) and send it to CS for pre-diagnosis services. After receiving the returned information, PT can read the pre-diagnostic result;
- CS is considered to have powerful computing ability and abundant storage space, and it undertakes the main calculations for pre-diagnosis. Specifically, on receiving a trapdoor, CS will search the encrypted tree and find out the top-k medical records with the nearest Mahalanobis distances to the query. Finally, the corresponding encrypted disease labels will be sent to PT.

B. Threat Model

In our system model, we assume that CS is *honest-but-curious*, which means that CS will strictly follow the proposed scheme but try to analyze the ciphertexts to obtain the original information as much as possible. Meanwhile, HI and PTs are considered to be trustable, i.e., HI will honestly provide registration services to PTs and outsource encrypted data to CS; PTs will send correct trapdoors to CS. Neither PTs nor HI will collude with CS. Besides, there exists a secure communication channel without external attacks.

In terms of the protection of sensitive information, we consider known-background attack (KBA) [18] in our threat model and its definition is given as follows.

Definition 1 (Known-background attack (KBA)). Apart from accessing ciphertexts, the attacker is also equipped with additional background information, such as disease types and statistical information. It can infer the specific contents in a query and database based on the extra information, but corresponding ciphertext and indexes are unknown. In other words, the attacker

TABLE I
DEFINITION OF MAIN NOTATIONS

Notations	Definition
N	The number of medical records.
n	The dimension of physiological data.
$\vec{q}$, $\vec{s}_i$	n-dimensional request and records.
Σ_*	The covariance matrix of a cluster, and * is the wildcard symbol.
$\mathcal{M}, \mathcal{M}_{0,\ldots,n}$	All are $(n+2)\times(n+2)$-dimensional random invertible matrices chosen as private key.
Γ	A random number chosen as public key.
$\vec{e}$	A random integer noise vector.
K	The number of clusters.
$\mathcal{T}$	A hierarchical index tree.
$\boldsymbol{C}_{h,d}(h=0)$	The leaf nodes of $\mathcal{T}$.
$\boldsymbol{C}_{h,d}(h>0)$	The non-leaf nodes of $\mathcal{T}$.
$C_{h,d}$	The cluster stored in $\boldsymbol{C}_{h,d}$.
$\vec{c}_{h,d}$	The center point of $C_{h,d}$.
$E(\cdot)$	The encrypted form of request, records, nodes, tree and so on.
ρ	The scale factor used in our schemes.
DT	a disease index table.

can obtain some plaintext and ciphertext, but cannot obtain the plaintext-ciphertext pair.

More formally, assuming that there exists a database $\boldsymbol{S} = \{\vec{s}_i|_{i=1}^N\}$ and its encrypted form $E(\boldsymbol{S}) = \{E(\vec{s}_i)|_{i=1}^N\}$, the attacker can obtain $\forall E(\vec{s}_i) \in E(\boldsymbol{S})$ and several $\vec{s}_j \in \boldsymbol{S}$, but it has no way of knowing the relationship between i and j, i.e., it cannot find the ciphertext of $\vec{s}_j$ from $E(\boldsymbol{S})$.

C. Privacy Requirements

Under the aforementioned system and threat models, the following requirements should be met simultaneously to provide privacy-preserving medical pre-diagnosis services.

- *Data Confidentiality.* Medical records collected by HI are private asset, and physiological data sent by PTs is sensitive information, both of them should be accessed only by the owners. Thus, the medical records and query requests must be kept secret from CS.
- *Pre-diagnosis privacy.* The pre-diagnostic result reveals the current health status of a patient, which PT does not want to disclose to others. Therefore, the proposed scheme should guarantee that the result can only be achieved by a legal PT.
- *Trapdoor unlinkability.* CS can observe the trapdoor in each query. To prevent the exposure of potential information, it should be ensured that CS cannot deduce the relationship of the received trapdoors or distinguish their difference, even the trapdoors are generated by the same query.

III. A SECURE SIMILARITY COMPARISON METHOD BASED ON MAHALANOBIS DISTANCE

In this section, we first review Mahalanobis distance [19] and the matrix encryption technique [20], and then design a secure similarity comparison method based on them. The key notations used in this paper are listed in Table I.

A. Mahalanobis Distance

Given a vector $\vec{q} = (q_1, \ldots, q_n) \in \mathbb{S}^n$ and a sample database $\boldsymbol{S} = \{\vec{s}_1, \ldots, \vec{s}_N\}$, where $\vec{s}_i = (s_{i1}, \ldots, s_{in}) \in \mathbb{S}^n, 1 \le i \le N$. Then, the ω-dimensional $(1 \le \omega \le n)$ mean of $\boldsymbol{S}$ can be computed as

$$\overline{d}_\omega = \frac{1}{N}(s_{1\omega} + s_{2\omega} + \cdots + s_{N\omega}).$$

Based on the average values, each element $\Sigma_{\boldsymbol{S}}(a,b)$ in the $n \times n$ covariance matrix of $\boldsymbol{S}$ can be obtained by

$$\Sigma_{\boldsymbol{S}}(a,b) = \frac{1}{N-1}\sum_{i=1}^{N}(x_{ia} - \overline{d}_a)(x_{ib} - \overline{d}_b),$$

where $a, b \in [1, n]$, and $\Sigma_{\boldsymbol{S}}(a,b)$ represents the element in the ath row and bth column of $\Sigma_{\boldsymbol{S}}$. Obviously, $\Sigma_{\boldsymbol{S}}(a,b) = \Sigma_{\boldsymbol{S}}(b,a)$, and $\Sigma_{\boldsymbol{S}}$ is a symmetric matrix.

After forming the covariance matrix $\Sigma_{\boldsymbol{S}}$, the Mahalanobis distance (MD) between $\vec{q}$ and $\vec{s}_i$ can be calculated by the following equation

$$\mathrm{MD}(\vec{q}, \vec{s}_i) = \sqrt{(\vec{q} - \vec{s}_i)\Sigma_{\boldsymbol{S}}^{-1}(\vec{q} - \vec{s}_i)^T},$$

where $\Sigma_{\boldsymbol{S}}^{-1}$ is the inverse of $\Sigma_{\boldsymbol{S}}$.

If we want to compute the Mahalanobis distance between $\vec{q}$ and the database $\boldsymbol{S}$, we can first calculate the center point of $\boldsymbol{S}$ as

$$\vec{c}_{\boldsymbol{S}} = (\overline{d}_1, \overline{d}_1, \ldots, \overline{d}_n),$$

then the distance can be computed by $\mathrm{MD}(\vec{q}, \vec{c}_{\boldsymbol{S}})$.

B. A Matrix Encryption Technique

The matrix encryption technique proposed by Yuan et al. named METY, can achieve euclidean distance comparison over ciphertexts [20]. Assuming there exists N scaled data records $\vec{s}_i = (s_{i1}, s_{i2}, \ldots, s_{in}) \in \mathbb{Z}_p^n, i \in [1, N]$, and a scaled comparison request $\vec{q} = (q_1, q_2, \ldots, q_n) \in \mathbb{Z}_p^n$ (all elements are scaled to integers with the same scale factor). The privacy-preserving distance comparison from the request to all data records can be achieved by the following algorithms.

KeyGen(n): The algorithm randomly selects an invertible matrices $\mathcal{M} \in \mathbb{R}_q^{2n\times 2n}$ and a number $\Gamma \in \mathbb{Z}_q$, $q \gg p$. Then it computes the inverse of $\mathcal{M}$, and sets $sk = \{\mathcal{M}, \mathcal{M}^{-1}\}$ as the private key, $pk = \Gamma$ as the public key.

DataEnc$(\vec{s}_i, sk, pk)$: At first, the algorithm expands each data record as

$$\vec{S}_i = \left(s_{i1}, \ldots, s_{in}, -\frac{1}{2}\sum_{j=1}^{n} s_{ij}^2, \beta_1, \ldots, \beta_{n-1}\right), i \in [1, N],$$

where $\beta_1, \ldots, \beta_{n-1}$ are random numbers chosen from $\mathbb{Z}_p$. Then, based on the extended vector, $\vec{s}_i$ is encrypted into

$$E(\vec{s}_i) = (\Gamma \cdot \vec{S}_i + \vec{e}_i) \times \mathcal{M},$$

where $\vec{e}_i \in \mathbb{Z}_q^{2n}$ is a random integer noise vector generated for each $\vec{s}_i$, and it satisfies $2 \cdot |max(\vec{e}_i)|$[1] $\ll \Gamma$.

1. $|\max(\cdot)|$ is defined to be the maximum absolute in a vector or a matrix.

TrapGen($\vec{q}$, sk, pk): At first, the algorithm expands the comparison request as

$$\vec{Q} = (rq_1, \ldots, rq_n, r, \alpha_1, \ldots, \alpha_{n-1}),$$

where $r, \alpha_1, \ldots, \alpha_{n-1}$ are random numbers chosen from $\mathbb{Z}_p$, and r is positive. Then, based on the extended vector, $\vec{q}$ is encrypted into

$$E(\vec{q}) = \mathcal{M}^{-1} \times (\Gamma \cdot \vec{Q}^T + \vec{e}_q^T),$$

where $\vec{e}_q \in \mathbb{Z}_q^{2n}$ is a random integer noise vector and satisfies $2 \cdot |max(\vec{e}_q)| \ll \Gamma$.

DisComp($E(\vec{q})$, $E(\vec{s}_{i|i\in\{a,b\}})$, pk): Given the encrypted comparison request $E(\vec{q})$ and any two encrypted data records $E(\vec{s}_a)$, $E(\vec{s}_b)$, the algorithm computes

$$Comp_{i|i\in\{a,b\}} = \left\lceil \frac{E(\vec{q}) \times E(\vec{s}_i)}{\Gamma^2} \right\rfloor_q,$$

where $\lceil \cdot \rfloor_q$ denotes the nearest integer with modulus q. The larger the value of $Comp_i$ is, the closer $\vec{s}_i$ is to $\vec{q}$.

In [20], the authors prove the security of METY based on the hardness assumption of the Learning With Error (LWE) problem [21]. However, the parameters defined in METY cannot meet the distribution requirements of standard LWE. To ensure that METY is secure under KBA, i.e., the original data $\vec{s}_{i|i\in[1,N]}$ and $\vec{q}$ can be protected, we formally analyze the security of METY in Section V.

C. Design of a Secure Similarity Comparison Method

In this subsection, by tailoring the above-mentioned matrix encryption technique, we design a secure similarity comparison method based on Mahalanobis distance called SMDC. Specifically, assuming that there exists two scaled data records $\vec{s}_a, \vec{s}_b \in \mathbb{Z}_p^n$ and one scaled comparison request $\vec{q} = (q_1, \ldots, q_n) \in \mathbb{Z}_p^n$, both $\vec{s}_a$ and $\vec{s}_b$ are selected from database $\boldsymbol{S}$ whose covariance matrix is $\Sigma_{\boldsymbol{S}}$. In order to compare the values of MD($\vec{q}, \vec{s}_a$) and MD($\vec{q}, \vec{s}_b$), the scaled inverse of $\Sigma_{\boldsymbol{S}}$ should be computed at first, i.e., $\Sigma_{\boldsymbol{S}}^{-1} \in \mathbb{Z}_p^{n\times n}$. Note that, all elements are scaled into integers with the same scale factor. Then, the following four algorithms, denoted as *KeyGen*, *DataEnc*, *TrapGen* and *DisComp*, are executed.

KeyGen(n): The algorithm inputs the dimension of data records n. It can output a $(n+2) \times (n+2)$-dimensional random invertible matrix $\mathcal{M} \in \mathbb{Z}_q^{(n+2)\times(n+2)}$ as the private key sk, and a random large number $\Gamma \in \mathbb{Z}_q, q \gg p$ as the public key pk.

DataEnc($\vec{s}_{i|i\in\{a,b\}}$, $\Sigma_{\boldsymbol{S}}^{-1}$, sk, pk): The algorithm inputs data records $\vec{s}_i$, the inverse of covariance matrix $\Sigma_{\boldsymbol{S}}^{-1}$, private key sk and public key pk. It can output the encrypted records $E(\vec{s}_a)$ and $E(\vec{s}_b)$ as shown in Algorithm 1. For each record $\vec{s}_i$, it first calculates

$$S_i = \vec{s}_i \Sigma_{\boldsymbol{S}}^{-1} \vec{s}_i^T,$$

$$\tilde{\boldsymbol{s}}_i = \Sigma_{\boldsymbol{S}}^{-1} \vec{s}_i^T = (\overline{s}_{i1}, \overline{s}_{i2}, \ldots, \overline{s}_{in})^T,$$

and extends $\tilde{\boldsymbol{s}}_i$ to a $(n+2)$-dimensional column vector as

$$\vec{S}_i = (-2\overline{s}_{i1}, -2\overline{s}_{i2}, \ldots, -2\overline{s}_{in}, S_i, \alpha)^T,$$

Algorithm 1. SMDC.DataEnc.

Input: Data records $\vec{s}_{i|1\le i\le N}$ in database $\boldsymbol{S}$, the inverse of $\boldsymbol{S}$'s covariance matrix $\Sigma_{\boldsymbol{S}}^{-1}$, private key $\mathcal{M}$, public key Γ.
Output: Encrypted records $E(\vec{s}_{i|1\le i\le N})$.

```
Randomly choose a number α ∈ Z_p;
for 1 ≤ i ≤ N do
    S_i = s_i · Σ_S^{-1} · s_i^T;
    s̃_i = Σ_S^{-1} s_i^T;
    Initialize a (n+2)-dimensional column vector S_i;
    for 0 ≤ x ≤ n − 1 do
        S_i[x] = −2 · s̃_i[x];
    S_i[n] = S_i;
    S_i[n+1] = α;
    Randomly choose a integer noise vector e_i ∈ Z_q^{n+2};
    E(s_i) = M^{-1} × (Γ · S_i + e_i^T);
return Encrypted records E(s_{i|1≤i≤N}).
```

where $\alpha \in \mathbb{Z}_p$ is a random number. After that, $\vec{s}_i$ is encrypted into $E(\vec{s}_i)$ by executing

$$E(\vec{s}_i) = \mathcal{M}^{-1} \times (\Gamma \cdot \vec{S}_i + \vec{e}_i^T),$$

where $\vec{e}_i \in \mathbb{Z}_q^{n+2}$ is a random integer noise vector generated for each record $\vec{s}_i$, $|\max(\vec{e}_i)| \ll \Gamma/2$; and the vector generated from $\mathcal{M}^{-1} \times \vec{e}_i^T$ satisfies a data distribution.

TrapGen($\vec{q}$, sk, pk): The algorithm inputs the comparison request $\vec{q}$, private key sk and public key pk. It can output a trapdoor $E(\vec{q})$ as shown in Algorithm 2. Specifically, it first extends $\vec{q}$ to a $(n+2)$-dimensional vector as

$$\vec{Q} = (rq_1, rq_2, \ldots, rq_n, r, \beta),$$

where $r \in \mathbb{Z}_p, \beta \in \mathbb{Z}_p$ are random numbers, and r is a positive integer. Then, it encrypts $\vec{q}$ into $E(\vec{q})$ as

$$E(\vec{q}) = (\Gamma \cdot \vec{Q} + \vec{e}_q) \times \mathcal{M},$$

where $\vec{e}_q \in \mathbb{Z}_q^{n+2}$ ($|\max(\vec{e}_q)| \ll \Gamma/2$) is also a random integer noise vector, and the vector generated from $\vec{e}_q \times \mathcal{M}$ also satisfies a data distribution.

DisComp($E(\vec{s}_{i|i\in\{a,b\}})$, $E(\vec{q})$, pk): The algorithm inputs the encrypted records $E(\vec{s}_a), E(\vec{s}_b)$, the trapdoor $E(\vec{q})$ and the public key pk. It can compare the values of MD($\vec{q}, \vec{s}_a$) and MD($\vec{q}, \vec{s}_b$) as shown in Algorithm 3. Specifically, it first calculates $D_i, i \in \{a, b\}$ by

$$D_i = \left\lceil \frac{E(\vec{q}) \times E(\vec{s}_i)}{\Gamma^2} \right\rfloor_q.$$

Then, it judges whether $D_a \le D_b$, if it does, MD($\vec{q}, \vec{s}_a$) $\le$ MD($\vec{q}, \vec{s}_b$) holds; otherwise, MD($\vec{q}, \vec{s}_a$) $>$ MD($\vec{q}, \vec{s}_b$).

Algorithm 2. SMDC.TrapGen.

Input: Request vector $\vec{q}$, private key $\mathcal{M}$, public key Γ.
Output: Trapdoor $E(\vec{q})$.

Initialize a $(n+2)$-dimensional vector $\vec{Q}$;
Randomly choose a number $r \in \mathbb{Z}_p$;
for $0 \leq x \leq n-1$ **do**
  $\vec{Q}[x] = r \cdot \vec{q}[x]$;
$\vec{Q}[n] = r$;
Randomly choose a number $\beta \in \mathbb{Z}_p$;
$\vec{Q}[n+1] = \beta$;
Randomly choose a integer noise vector $\vec{e}_q \in \mathbb{Z}_q^{n+2}$;
$E(\vec{q}) = (\Gamma \cdot \vec{Q} + \vec{e}_q) \times \mathcal{M}$;
return Trapdoor $E(\vec{q})$.

Algorithm 3. SMDC.*DisComp*.

Input: Encrypted records $E(\vec{s}_{i|1\leq i\leq N})$, the trapdoor $E(\vec{q})$, public key Γ.
Output: The ascending order of $\{\mathrm{MD}(\vec{q},\vec{s}_i)|1 \leq i \leq N\}$.

for $1 \leq i \leq N$ **do**
  $D_i = \left\lceil \frac{E(\vec{q}) \times E(\vec{s}_i)}{\Gamma^2} \right\rfloor_q$;
  $\mathrm{MD}(\vec{q},\vec{s}_i) = D_i$;
Sort $\{\mathrm{MD}(\vec{q},\vec{s}_1), \ldots, \mathrm{MD}(\vec{q},\vec{s}_N)\}$ in ascending order;
return The ascending order of $\{\mathrm{MD}(\vec{q},\vec{s}_i)|1 \leq i \leq N\}$.

Correctness of SMDC. Obviously, the correctness of SMDC is decided by the establishment of $D_i \leq D_j \Rightarrow \mathrm{MD}(\vec{q},\vec{s}_a) \leq \mathrm{MD}(\vec{q},\vec{s}_b)$.

Proof. Firstly, based on $E(\vec{q}) = (\Gamma \cdot \vec{Q} + \vec{e}_q) \times \mathcal{M}$ and $E(\vec{s}_i) = \mathcal{M}^{-1} \times (\Gamma \cdot \vec{S}_i + \vec{e}_i^T), i \in \{a, b\}$, we expand D_i as

$$D_i = \left\lceil \frac{\Gamma^2 \cdot (\vec{Q} \times \vec{S}_i) + \Gamma \cdot (\vec{Q} \times \vec{e}_i^T + \vec{e}_q \times \vec{S}_i) + \vec{e}_q \times \vec{e}_i^T}{\Gamma^2} \right\rfloor_q$$

$$= rS_i - 2 \cdot \sum_{j=1}^{n} rq_j\overline{s}_{ij} + \alpha\beta$$

$$= r \cdot \left(\Omega - 2 \cdot \sum_{j=1}^{n} q_j\overline{s}_{ij} + S_i \right) - r \cdot \Omega + \alpha\beta.$$

Then, assuming $\Omega = \vec{q}\Sigma_{\boldsymbol{S}}^{-1}\vec{q}^T$, $\Theta = -r \cdot \Omega + \alpha\beta$, D_i can be transformed into

$$D_i = r\left(\vec{q}\Sigma_{\boldsymbol{S}}^{-1}\vec{q}^T - 2\vec{q}\Sigma_{\boldsymbol{S}}^{-1}\vec{s}_i^T + \vec{s}_i\Sigma_{\boldsymbol{S}}^{-1}\vec{s}_i^T\right) + \Theta.$$

Due to $\Sigma_{\boldsymbol{S}}$ is a symmetric matrix, $\Sigma_{\boldsymbol{S}}^{-1}$ must be another symmetric matrix, we can deduce that $\vec{q}\Sigma_{\boldsymbol{S}}^{-1}\vec{s}_i^T = \vec{s}_i\Sigma_{\boldsymbol{S}}^{-1}\vec{q}^T$. Therefore,

$$\begin{aligned} D_i &= r \cdot \left(\vec{q}\Sigma_{\boldsymbol{S}}^{-1}\vec{q}^T - 2\vec{q}\Sigma_{\boldsymbol{S}}^{-1}\vec{s}_i^T + \vec{s}_i\Sigma_{\boldsymbol{S}}^{-1}\vec{s}_i^T\right) + \Theta \\ &= r \cdot \left(\vec{q}\Sigma_{\boldsymbol{S}}^{-1}\vec{q}^T - \vec{s}_i\Sigma_{\boldsymbol{S}}^{-1}\vec{q}^T - \vec{q}\Sigma_{\boldsymbol{S}}^{-1}\vec{s}_i^T + \vec{s}_i\Sigma_{\boldsymbol{S}}^{-1}\vec{s}_i^T\right) + \Theta \\ &= r \cdot (\vec{q} - \vec{s}_i)\,\Sigma_{\boldsymbol{S}}^{-1}\vec{q}^T - r \cdot (\vec{q} - \vec{s}_i)\,\Sigma_{\boldsymbol{S}}^{-1}\vec{s}_i^T + \Theta \\ &= r \cdot (\vec{q} - \vec{s}_i)\Sigma_{\boldsymbol{S}}^{-1}(\vec{q} - \vec{s}_i)^T + \Theta \\ &= r \cdot \mathrm{MD}(\vec{q},\vec{s}_i) + \Theta. \end{aligned}$$

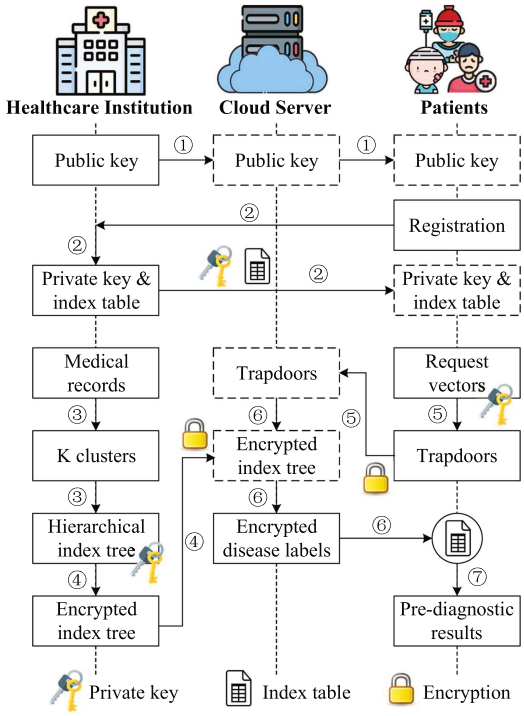


Fig. 2. Overview of NAIAD.

Based on the above equations, the difference of $\mathrm{MD}(\vec{q},\vec{s}_a)$ and $\mathrm{MD}(\vec{q},\vec{s}_b)$ can be easily obtained through

$$D_a - D_b = r \cdot [\mathrm{MD}(\vec{q},\vec{s}_a) - \mathrm{MD}(\vec{q},\vec{s}_b)].$$

As r is a positive integer, it is clear that the comparison of encrypted distances (i.e., D_a, D_b) is consistent with the comparison of exact distances (i.e., $\mathrm{MD}(\vec{q},\vec{s}_a), \mathrm{MD}(\vec{q},\vec{s}_b)$).

Remark. The correctness of the equations deduced in *Proof* is guaranteed by the properties of matrix operations, the same scale factor, and the fact that $\Gamma \gg p$, $\Gamma \gg 2 \cdot |\max(\vec{e}_i)|$ and $\Gamma \gg 2 \cdot |\max(\vec{e}_q)|$. □

IV. An Efficient and Privacy-Preserving Online Pre-Diagnosis Scheme

In this section, an efficient and accurate cloud-assisted medical pre-diagnosis scheme, named NAIAD, is proposed based on SMDC. Specifically, NAIAD consists of four main phases, namely, *1) system initialization*, *2) encrypted index tree outsourcing*, *3) trapdoor generation*, and *4) privacy-preserving online pre-diagnosis*. The overview of NAIAD is described in Fig. 2. HI first generates the system parameters (①②), where the public key is made publicly, and the private key is shared with registered PTs. Next, it partitions the scaled medical records into K clusters, and builds a hierarchical index tree over the clusters (③). Then, the hierarchical index tree would be encrypted and sent to CS (④). After that, PTs encrypt their query requests (i.e., physiological data) to generate trapdoors which need to be sent to CS for pre-diagnosis (⑤), and CS will search the encrypted tree to return the top-k similar medical records' encrypted disease

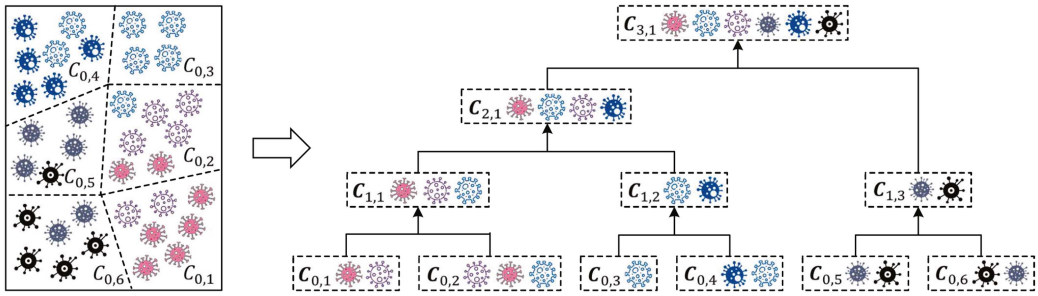


Fig. 3. Construction of hierarchical index tree.

labels to PTs (⑥). Finally, PTs can obtain the pre-diagnostic results by decrypting the received encrypted labels (⑦).

A. System Initialization

In this phase, HI first generates system parameters and provides registration services to PTs, then it constructs a hierarchical index tree over the stored medical records.

Assuming that the dimension of patients' physiological data is n, HI randomly chooses $n+1$ invertible matrices $\{\mathcal{M}_0, \ldots, \mathcal{M}_n\} \in \mathbb{Z}_q^{(n+2)\times(n+2)}$, n integers $\{\delta_1, \ldots, \delta_n\} \in \mathbb{Z}_p$, a scale factor ρ, and a large number $\Gamma \in \mathbb{Z}_q, q \gg p$. It also generates a key pair (pk_E, sk_E) for a public-key encryption (PKE) algorithm. Finally, HI publishes PK= $\{\Gamma, pk_E\}$ and keeps SK= $\{\mathcal{M}_0, \ldots, \mathcal{M}_n, \delta_1, \ldots, \delta_n, \rho, sk_E\}$ secret.

When PT registers at HI, it first needs to submit its identification information to HI through a secure communication channel. Then, if PT is considered to be legal, it will receive SK= $\{\mathcal{M}_0, \ldots, \mathcal{M}_n, \delta_1, \ldots, \delta_n, \rho, sk_E\}$, and a disease index table DT from HI; otherwise the patient is not allowed to access the provided pre-diagnosis service.

Besides, in a bottom-up manner, a hierarchical index tree $\mathcal{T}$ is built by HI in this phase to speed up the query efficiency. The core idea of construction is to group similar medical records into a cluster, and then group similar clusters into higher ones. We denote each node of $\mathcal{T}$ as $\boldsymbol{C}_{h,d} = \{C_{h,d}, \Sigma_{h,d}^{-1}, \vec{c}_{h,d}\}$, where h is the located level, d is the index at level h, $C_{h,d}$ represents the cluster stored in $\boldsymbol{C}_{h,d}$, $\Sigma_{h,d}^{-1}$ represents the inverse of $C_{h,d}$'s covariance matrix and $\vec{c}_{h,d}$ represents the center point of $C_{h,d}$. After HI divides the stored medical records into K clusters through an existing clustering algorithm which based on Mahalanobis distance [22], [23], the K clusters can be regarded as the leaf nodes of $\mathcal{T}$, denoted as $\{\boldsymbol{C}_{0,d} = (C_{0,d}, \Sigma_{0,d}^{-1}, \vec{c}_{0,d}) | d \in [1, K]\}$. Based on the K center points $\vec{c}_{0,d}$, the Mahalanobis distance (MD) between any two leaf nodes can be calculated. Then, the most similar two clusters with the closest distance will be merged into a new cluster, and their parent node will be formed by calculating the canter point and the inverse of the covariance matrix of the new cluster. Finally, $\mathcal{T}$ is constructed iteratively until all leaf nodes are clustered into one root node, and all elements in $\mathcal{T}$ are scaled into integers with ρ.

To clearly express the construction process, an example is given in Fig. 3. Specifically, in the given example, the hierarchical index tree is constructed by recursively merging the two closest clusters (the number of the merged clusters can be any number in theoretical). Since the MD between $C_{0,1}$ and $C_{0,2}$ in the set $\{C_{0,1}, C_{0,2}, \ldots, C_{0,6}\}$ is the closest, $C_{0,1}$ and $C_{0,2}$ are first merged into $C_{1,1} = C_{0,1} \cup C_{0,2}$ to form a new mode $\boldsymbol{C}_{1,1}$. Then, based on the updated set $\{C_{1,1}, C_{0,3}, \ldots, C_{0,6}\}$, $C_{1,2} = C_{0,3} \cup C_{0,4}$ can be obtained by calculating the closest MD again. Recursively, $C_{1,3} = C_{0,5} \cup C_{0,6}$, $C_{2,1} = C_{1,1} \cup C_{1,2}$, and $C_{3,1} = C_{2,1} \cup C_{1,3}$ can be generated in sequence.

Algorithm 4. NAIAD.NodeEnc.

Input: The non-leaf node $\boldsymbol{C}_{h,d} = \{\Sigma_{h,d}^{-1}, \vec{c}_{h,d}\}$, private key $\{\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_n, \delta_1, \ldots, \delta_n\}$, public key Γ, and random numbers $\{\mu_{\omega 1}, \mu_{\omega 2}\}_{1 \le \omega \le n}$.
Output: Encrypted non-leaf node $E(\boldsymbol{C}_{h,d})$.

```
Extract each column σ⃗_1, ..., σ⃗_n from Σ_{h,d}^{-1};
for 1 ≤ ω ≤ n do
    Initialize a n+2-dimensional column vector Ô_ω;
    for 0 ≤ i ≤ n−1 do
        Ô_ω[i] = δ_ω · σ⃗_ω[i]
    Ô_ω[n] = μ_{ω1};
    Ô_ω[n+1] = μ_{ω2};
    Randomly choose a integer noise vector e⃗_ω ∈ Z_q^{n+2};
    E(σ⃗_ω) = M_ω^{-1} × (Γ · Ô_ω + e⃗_ω^T);
E(c⃗_{h,d})=SMDC.DataEnc(c⃗_{h,d}, Σ_{h,d}^{-1}, M_0, Γ);
E(C_{h,d}) = {E(σ⃗_ω)|_{ω=1}^n, E(c⃗_{h,d})};
return Encrypted non-leaf node E(C_{h,d}).
```

B. Encrypted Index Tree Outsourcing

In this phase, HI encrypts the hierarchical index tree, and outsources it to CS for providing PTs with medical pre-diagnosis services.

At first, HI updates the non-leaf ($h > 0$) nodes $\boldsymbol{C}_{h,d} = \{C_{h,d}, \Sigma_{h,d}^{-1}, \vec{c}_{h,d}\}$ in $\mathcal{T}$ by following the principle: if the node is a root node, replace it with $\varnothing$; otherwise, remove the cluster $C_{h,d}$, and replace it with $\{\Sigma_{h,d}^{-1}, \vec{c}_{h,d}\}$.

Then, HI encrypts the non-leaf nodes (except root node) and leaf nodes in two different methods.

• *Non-Leaf nodes Encryption*

For each non-leaf node $\boldsymbol{C}_{h,d} = \{\Sigma_{h,d}^{-1}, \vec{c}_{h,d}\}$ ($h > 0$), where $\Sigma_{h,d}^{-1} \in \mathbb{Z}_p^{n\times n}$, $\vec{c}_{h,d} \in \mathbb{Z}_p^n$, the encryption process is given in Algorithm 4.

Specifically, HI first extracts each column of matrix $\Sigma_{h,d}^{-1}$, and denotes the ωth column as $\vec{\sigma}_\omega = (\sigma_{1\omega}, \sigma_{2\omega}, \ldots, \sigma_{n\omega})^T, 1 \leq \omega \leq n$. Next, HI extends $\vec{\sigma}_\omega$ to $(n+2)$-dimensional as

$$\vec{O}_\omega = (\delta_\omega \cdot \sigma_{1\omega}, \ldots, \delta_\omega \cdot \sigma_{n\omega}, \mu_{\omega 1}, \mu_{\omega 2})^T,$$

where $\mu_{\omega 1} \in \mathbb{Z}_p, \mu_{\omega 2} \in \mathbb{Z}_p$ are random numbers.

Then, with the invertible matrix $\mathcal{M}_\omega$, public key Γ, and a random integer noise vector $\vec{e}_\omega \in \mathbb{Z}_q^{n+2}$ ($|\max(\vec{e}_\omega)| \ll \Gamma/(n+2)$), $\vec{\sigma}_\omega$ can be encrypted into

$$E(\vec{\sigma}_\omega) = \mathcal{M}_\omega^{-1} \times (\Gamma \cdot \vec{O}_\omega + \vec{e}_\omega^T).$$

After that, HI runs SMDC.*DataEnc* to encrypt $\vec{c}_{h,d}$. Specifically, it computes $\tilde{\boldsymbol{c}}_{h,d} = \Sigma_{h,d}^{-1}\vec{c}_{h,d}^T = (\bar{c}_{h,d}^{(1)}, \ldots, \bar{c}_{h,d}^{(n)})^T$, and extends $\tilde{\boldsymbol{c}}_{h,d}$ to $(n+2)$-dimensional as

$$\vec{C}_{h,d} = (-2 \cdot \bar{c}_{h,d}^{(1)}, \ldots, -2 \cdot \bar{c}_{h,d}^{(n)}, \vec{c}_{h,d}\Sigma_{h,d}^{-1}\vec{c}_{h,d}^T, \alpha'),$$

where random number $\alpha' \in \mathbb{Z}_p$ are same for different center points.

Next, $\vec{c}_{h,d}$ is encrypted into

$$E(\vec{c}_{h,d}) = \mathcal{M}_0^{-1} \times (\Gamma \cdot \vec{C}_{h,d} + \vec{e}_{h,d}^T),$$

where $\vec{e}_{h,d} \in \mathbb{Z}_q^{n+2}$ is an integer noise vector, and satisfies $|\max(\vec{e}_{h,d})| \ll \Gamma/(n+2)$.

As a result, each non-leaf node $\boldsymbol{C}_{h,d}$ (except the root node) is encrypted into $E(\boldsymbol{C}_{h,d}) = \{E(\vec{\sigma}_\omega)|_{\omega=1}^n, E(\vec{c}_{h,d})\}$.

• *Leaf nodes Encryption*

For each leaf node $\boldsymbol{C}_{0,d} = \{C_{0,d}, \Sigma_{0,d}^{-1}, \vec{c}_{0,d}\}$ $(0 \leq d \leq K)$, where $C_{0,d} = \{\vec{s}_{d1}, \ldots, \vec{s}_{di}, \ldots, \vec{s}_{dl_d}\}$, each $\vec{s}_{di} \in \mathbb{Z}_p^n$ is a uniform vector and l_d counts the number of medical records in $C_{0,d}$, HI first encrypts $\{\Sigma_{0,d}^{-1}, \vec{c}_{0,d}\}$ into $\{E(\vec{\sigma}_\omega)|_{\omega=1}^n, E(\vec{c}_{0,d})\}$ by running NAIAD.*NodeEnc*. Then, for each $\vec{s}_{di}$ stored in $C_{0,d}$, HI runs SMDC.*DataEnc*$(\vec{s}_{di}, \Sigma_{0,d}^{-1}, \mathcal{M}_0, \Gamma)$ to encrypt it and obtains $E(\vec{s}_{di})$.

Finally, the leaf node $\boldsymbol{C}_{0,d}$ can be encrypted into $E(\boldsymbol{C}_{0,d})$, which can be denoted as $\{E(\vec{s}_{di})|_{i=1}^{l_d}, E(\vec{\sigma}_\omega)|_{\omega=1}^n, E(\vec{c}_{0,d})\}$.

After encrypting all nodes of $\mathcal{T}$ (except the root node), HI outsources the encrypted index tree $E(\mathcal{T})$ to CS. Notably, each $\vec{s}_{di}$ is associated with a disease label, and each label is encrypted by PKE with pk_E.

C. Trapdoor Generation

In this phase, PT encrypts its query request and sends the trapdoor to CS.

At first, the registered PT scales its request vector into $\vec{q} = (q_1, q_2, \ldots, q_n) \in \mathbb{Z}_p^n$ with ρ, then it obtain $E(\vec{q})$ by running SMDC.*TrapGen*$(\vec{q}, \mathcal{M}_0, \Gamma)$, during the encryption process, the chosen $r = c \cdot \delta_1\delta_2 \cdots \delta_n$, c is an random positive integer, and the integer noise vector satisfies $|\max(\vec{e}_q)| \ll \Gamma/(n+2)$.

After that, PT multiplies $\vec{q}$ by each q_ω, $1 \leq \omega \leq n$ to obtain

$$\vec{\gamma}_\omega = q_\omega \cdot \vec{q} = (q_\omega \cdot q_1, \ldots, q_\omega \cdot q_n) = (\gamma_{\omega 1}, \ldots, \gamma_{\omega n}),$$

and extends it to

$$\vec{\Upsilon}_\omega = \left(\frac{r}{\delta_\omega} \cdot \gamma_{\omega 1}, \ldots, \frac{r}{\delta_\omega} \cdot \gamma_{\omega n}, \nu_{\omega 1}, \nu_{\omega 2}\right),$$

where $\nu_{\omega 1} \in \mathbb{Z}_p, \nu_{\omega 2} \in \mathbb{Z}_p$ are random numbers.

Next, $\vec{\gamma}_\omega$ can be encrypted by

$$E(\vec{\gamma}_\omega) = (\Gamma \cdot \vec{\Upsilon}_\omega + \vec{e}_\omega^*) \times \mathcal{M}_\omega,$$

where Γ is the public key, $\mathcal{M}_\omega$ are chosen from SK and $\vec{e}_\omega^*, 1 \leq \omega \leq n$ are different integer noise vectors randomly selected from $\mathbb{Z}_q^{n+2}$, satisfying $|\max(\vec{e}_\omega^*)| \ll \Gamma/(n+2)$.

Finally, PT sends the trapdoor TD$= \{E(\vec{\gamma}_\omega)|_{\omega=1}^n, E(\vec{q})\}$ to CS for online pre-diagnosis.

D. Privacy-Preserving Online Pre-Diagnosis

In this phase, CS first searches $E(\mathcal{T})$ to find the most similar cluster to the query request, then returns the top-k similar records' encrypted labels for result reading.

Upon receiving a trapdoor TD from PT, CS searches $E(\mathcal{T})$ from the root node $\varnothing$ as follows:

1) CS compares the Mahalanobis distances from the trapdoor to the encrypted center points stored in its two child nodes, set $E(\boldsymbol{C}_{h,d})$ to the more similar child node;
2) If $E(\boldsymbol{C}_{h,d})$ is a parent node, return to 1); otherwise, extracting the encrypted medical records from $E(\boldsymbol{C}_{h,d})$, where $h = 0$;
3) CS compares the Mahalanobis distances from the trapdoor to the encrypted medical records, find the top-k similar records, and output their corresponding encrypted labels.

Specifically, the similarity comparison in 1) is transformed into the comparison of $E_{h,d}$, which can be calculated by the following equations:

$$E_{h,d} = \left\lceil \frac{U_{h,d}}{\Gamma^2} \right\rfloor_q,$$

$$\text{where } U_{h,d} = \sum_{\omega=1}^{n} E(\vec{\gamma}_\omega) \cdot E(\vec{\sigma}_\omega) + E(\vec{q}) \cdot E(\vec{c}_{h,d}).$$

Meanwhile, the similarity comparison executed in 3) is achieved by SMDC.*DisComp*. Thus, CS can find the top-k medical records similar to the query request, and the corresponding k encrypted labels will be returned to PT.

Finally, PT decrypts the received encrypted labels via sk_E and obtains the diseases via disease labels and index table DT, then it can evaluate its health status.

Correctness of NAIAD. It can be observed that the correctness of NAIAD depends on two aspects: 1) the establishment of $E_{h_1,d_1} \leq E_{h_2,d_2} \Rightarrow \text{MD}(\vec{q}, \vec{c}_{h_1,d_1}) \leq \text{MD}(\vec{q}, \vec{c}_{h_2,d_2})$, where $\vec{c}_{h_1,d_1}, \vec{c}_{h_2,d_2}$ represent the center points stored in the two child nodes; 2) the correctness of SMDC. Due to 2) has been proven in Section III-C, here, we are devoted to proving 1). For ease of expression, h, d are considered as the universal characters of h_1, d_1 and h_2, d_2. The detailed derivation process is given as follows.

Proof. Firstly, expand $U_{h,d}$ as

$$\begin{aligned} U_{h,d} &= \sum_{\omega=1}^{n} E(\vec{\gamma}_\omega) \cdot E(\vec{\sigma}_\omega) + E(\vec{q}) \cdot E(\vec{c}_{h,d}) \\ &= \sum_{\omega=1}^{n} (\Gamma \cdot \vec{\Upsilon}_\omega + \vec{e}_\omega^*) \times \mathcal{M}_\omega \times \mathcal{M}_\omega^{-1} \times (\Gamma \cdot \vec{O}_\omega + \vec{e}_\omega^T) \end{aligned}$$

$$+ (\Gamma \cdot \vec{Q} + \vec{e}_q) \times \mathcal{M}_0 \times \mathcal{M}_0^{-1} \times (\Gamma \cdot \vec{C}_{h,d} + \vec{e}_{h,d}^T)$$

$$= \Gamma^2 \cdot \left(\vec{Q} \times \vec{C}_{h,d} + \sum_{\omega=1}^{n} \vec{\Upsilon}_\omega \times \vec{O}_\omega \right) + \Delta + \Psi,$$

$$\frac{\Delta}{\Gamma} = \vec{Q} \times \vec{e}_{h,d}^T + \vec{e}_q \times \vec{C}_{h,d} + \sum_{\omega=1}^{n} \left(\vec{\Upsilon}_\omega \times \vec{e}_\omega^T + \vec{e}_\omega^* \times \vec{O}_\omega \right),$$

$$\Psi = \vec{e}_q \times \vec{e}_{h,d}^T + \sum_{\omega=1}^{n} \vec{e}_\omega^* \times \vec{e}_\omega^T.$$

Then, due to $E_{h,d} = \lceil U_{h,d}/\Gamma^2 \rfloor_q$, $E_{h,d}$ can be calculated by

$$E_{h,d} = \vec{Q} \times \vec{C}_{h,d} + \sum_{\omega=1}^{n} \vec{\Upsilon}_\omega \times \vec{O}_\omega$$

$$= r \cdot \vec{c}_{h,d} \Sigma_{h,d}^{-1} \vec{c}_{h,d}^T - 2r \cdot \vec{q} \Sigma_{h,d}^{-1} \vec{c}_{h,d}^T + \alpha' \beta + \sum_{\omega=1}^{n} \left(\nu_{\omega 1} \mu_{\omega 1} + \nu_{\omega 2} \mu_{\omega 2} + \sum_{j=1}^{n} r \cdot q_\omega \cdot q_j \cdot \sigma_{j\omega} \right)$$

$$= r \cdot \left(\vec{q} \Sigma_{h,d}^{-1} \vec{q}^T - 2 \cdot \vec{q} \Sigma_{h,d}^{-1} \vec{c}_{h,d}^T + \vec{c}_{h,d} \Sigma_{h,d}^{-1} \vec{c}_{h,d}^T \right) + R$$

$$= r \cdot \mathrm{MD}(\vec{q}, \vec{c}_{h,d}) + R,$$

$$R = \alpha' \beta + \sum_{\omega=1}^{n} \left(\nu_{\omega 1} \mu_{\omega 1} + \nu_{\omega 2} \mu_{\omega 2} \right).$$

Obviously, the following equation is established

$$E_{h_1,d_1} - E_{h_2,d_2} = r \cdot \left[\mathrm{MD}(\vec{q}, \vec{c}_{h_1,d_1}) - \mathrm{MD}(\vec{q}, \vec{c}_{h_2,d_2}) \right].$$

Considering that $\Gamma \gg p$, $\Gamma \gg (n+2) \cdot |\max(\vec{e})|$, where $\vec{e} \in \{\vec{e}_\omega^*, \vec{e}_\omega, \vec{e}_q, \vec{e}_{h,d}\}$, and r is a positive integer, 1) is must correct. Consequently, the correctness of NAIAD is proved. □

V. Security Analysis

In this section, we first prove the security of METY and our designed basic comparison method SMDC, then we demonstrate that NAIAD can meet the privacy requirements listed in Section II-C.

To achieve the similarity comparison function on ciphertexts, both METY and SMDC have to leak the "closeness" and "equality" of encrypted distances, and the *Size Pattern*, *Access Pattern* and *Search Pattern* are assumed to be disclosed by default [24]. Therefore, we adopt an optimal security notion called *indistinguishability under closeness-same-pattern chosen-plaintext attack* (IND-CLS-CPA) for SMDC, which is a natural relaxation of the standard IND-CPA [25], [26]. IND-CLS-CPA is essentially equivalent to KBA, but the new notion can help conduct security analysis more formally, and it has been applied to many searchable encryption schemes [27], [28], [29].

In this section, a leakage function $\mathcal{L}(\boldsymbol{S}, \vec{q})$ is first defined to consider all possible leaks during the comparison (or query) process, then we prove that METY and SMDC can achieve IND-CLS-CPA record privacy and request privacy, finally the security of NAIAD is analyzed based on SMDC.

A. Leakage Function

Essentially, both METY and SMDC can be regarded as searchable encryption schemes. In the security definitions of searchable encryption, leakage function can be defined as follows.

Definition 2. (Leakage Function $\mathcal{L}(\boldsymbol{S}, \vec{q})$) Given a dataset $\boldsymbol{S}$ and a request $\vec{q}$, the leakage function $\mathcal{L}(\boldsymbol{S}, \vec{q})$ consists of three main parts: *1) Size Pattern*, the total number of data records in $\boldsymbol{S}$, the request times of patients and the dimensions of ciphertexts; *2) Access Pattern*, the identifier of each returned encrypted data record; *3) Search Pattern*, whether a comparison result is achieved by two trapdoors, even they are different.

B. Security Analysis of METY

Based on $\mathcal{L}(\boldsymbol{S}, \vec{q})$, we first give the security definitions of METY, and then analyze its security. Specifically, the privacy protection of original data records $\vec{s}_{i|i\in[1,N]}$ and comparison request $\vec{q}$ can be decomposed into *Record Privacy* and *Request Privacy*, and they strictly follow IND-CLS-CPA.

Definition 3. (IND-CLS-CPA Record Privacy) To capture the IND-CLS-CPA record privacy of METY, a security game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ is defined as:

Initial: $\mathcal{A}$ selects two scaled databases $\boldsymbol{S}_0 = \{\vec{s}_{0,1}, \vec{s}_{0,2}, \ldots, \vec{s}_{0,N}\}$, $\boldsymbol{S}_1 = \{\vec{s}_{1,1}, \vec{s}_{1,2}, \ldots, \vec{s}_{1,N}\}$, and $\{\boldsymbol{S}_b, \Sigma_{\boldsymbol{S}_b}^{-1}\}_{b\in\{0,1\}}$ are sent to $\mathcal{C}$.

Setup: $\mathcal{C}$ runs METY.*KeyGen*(1^λ) to generate private key sk, and public key pk.

Phase 1: $\mathcal{A}$ submits a number of adaptive chosen databases and requests to $\mathcal{C}$ for encryption:

- Database encryption (METY.*DataEnc*): On the jth database encryption, $\mathcal{A}$ selects a scaled dataset $\boldsymbol{S}'_j = \{\vec{s}'_{j,1}, \ldots, \vec{s}'_{j,N}\}$ and sends $\boldsymbol{S}'_j$ to $\mathcal{C}$. $\mathcal{C}$ will answer it with encrypted database $E(\boldsymbol{S}'_j)$ by METY.*DataEnc*($\boldsymbol{S}'_j$, sk, pk).
- Request encryption (METY.*TrapGen*): On the jth request encryption, $\mathcal{A}$ selects a scaled request $\vec{q}_j$ and sends it to $\mathcal{C}$. $\mathcal{C}$ will answer it with trapdoor $E(\vec{q}_j)$ by METY.*TrapGen*($\vec{q}_j$, sk, pk), where $\vec{q}_j$ should meet the following conditions simultaneously.
  1) $\mathcal{L}(\boldsymbol{S}_0, \vec{q}_j) = \mathcal{L}(\boldsymbol{S}_1, \vec{q}_j)$;
  2) For $1 \le i \le N$, $\vec{s}_{0,i}$ and $\vec{s}_{1,i}$ are included in the top-k records similar to $\vec{q}_j$ in the same order, or neither of them are in, where $1 \le k \le N$.

Challenge: With $\boldsymbol{S}_0$ and $\boldsymbol{S}_1$ received in phase *Initial*, $\mathcal{C}$ throws a coin to decide $b = 0$ or $b = 1$, then returns $E(\boldsymbol{S}_b)$ to $\mathcal{A}$ by METY.*DataEnc*($\boldsymbol{S}_b$, sk, pk).

Phase 2: Same as Phase 1, $\mathcal{A}$ continues to choose a number of adaptive databases and requests subjected to the same conditions, and submit them to $\mathcal{C}$ for responses.

Guess: $\mathcal{A}$ takes a guess b' of b.

Denote the advantage of any probabilistic polynomial time (PPT) adversary $\mathcal{A}$ in guessing $b' = b$ in the game above as $Adv_{\mathrm{METY},\mathcal{A}}^{\text{IND-CLS-CPA-Record}}$, METY is said to be IND-CLS-CPA record privacy *iff* the advantage is negligible.

Theorem 1. METY achieves IND-CLS-CPA record privacy under the above-defined game.

Proof. The security game defined in *Definition* 3 is exploited to analyze the security of METY.

Initial: $\mathcal{A}$ selects two scaled databases $\boldsymbol{S}_0 = \{\vec{s}_{0,1}, \vec{s}_{0,2}, \ldots, \vec{s}_{0,N}\}$, $\boldsymbol{S}_1 = \{\vec{s}_{1,1}, \vec{s}_{1,2}, \ldots, \vec{s}_{1,N}\}$, and $\{\boldsymbol{S}_b, \Sigma_{\boldsymbol{S}_b}^{-1}\}_{b\in\{0,1\}}$ are sent to $\mathcal{C}$, note that the dimension of each $\vec{s}_{b,i}$ is n.

Setup: $\mathcal{C}$ runs METY.*KeyGen*(1^λ) to generate private key $sk = \{\mathcal{M}, \mathcal{M}^{-1}\}$, and public key $pk = \Gamma$.

Phase 1: $\mathcal{A}$ submits a number of adaptive chosen databases and requests to $\mathcal{C}$ for encryption:

- Database encryption (METY.*DataEnc*): On the jth database encryption, $\mathcal{A}$ first selects a scaled database $\boldsymbol{S}'_j = \{\vec{s'}_{j,1}, \ldots, \vec{s'}_{j,N}\}$, and sends $\boldsymbol{S}'_j$ to $\mathcal{C}$. $\mathcal{C}$ will answer it with encrypted database $E(\boldsymbol{S}'_j) = \{E(\vec{s'}_{j,i})|_{i=1}^N\}$ by METY.*DataEnc*$(\boldsymbol{S}'_j, sk, pk)$, where each record $\vec{s'}_{j,i}$ in $\boldsymbol{S}'_j$ is encrypted to $E(\vec{s'}_{j,i}) = (\Gamma \cdot \vec{S'}_{j,i} + \vec{e'}_{j,i}) \times \mathcal{M}$, and each $\vec{S'}_{j,i} = (\underbrace{\vec{s'}_{j,i}^T}_{n}, -\frac{1}{2}\sum_{k=1}^n s_{ik}^2, \underbrace{\beta_1, \ldots, \beta_{n-1}}_{n-1})$, $\vec{e'}_{j,i}$ is a random integer noise vector.
- Request encryption (METY.*TrapGen*): On the jth request encryption, $\mathcal{A}$ selects a scaled n-dimensional request vector $\vec{q}_j$ and sends it to $\mathcal{C}$. $\mathcal{C}$ answers it with trapdoor $E(\vec{q}_j) = \mathcal{M}^{-1} \times (\Gamma \cdot \vec{Q}^T + \vec{e}_q^T)$ by running METY.*TrapGen*$(\vec{q}_j, sk, pk)$, where $\vec{Q} = (\underbrace{r \cdot \vec{q}_j}_{n}, r, \underbrace{\alpha_1, \ldots, \alpha_{n-1}}_{n-1})$, and $\vec{e}_q$ is a random integer noise vector. Besides, $\vec{q}_j$ also should meet the following conditions simultaneously.
  1) $\mathcal{L}(\boldsymbol{S}_0, \vec{q}_j) = \mathcal{L}(\boldsymbol{S}_1, \vec{q}_j)$;
  2) For $1 \le i \le N$, $\vec{s}_{0,i}$ and $\vec{s}_{1,i}$ are included in the top-k records similar to $\vec{q}_j$ in the same order, or neither of them are in, where $1 \le k \le N$.

Challenge: With $\boldsymbol{S}_0$ and $\boldsymbol{S}_1$ received in phase *Initial*, $\mathcal{C}$ throws a coin to decide $b = 0$ or $b = 1$, then returns $E(\boldsymbol{S}_b)$ to $\mathcal{A}$ by METY.*DataEnc*$(\boldsymbol{S}_b, sk, pk)$.

Phase 2: Same as Phase 1, $\mathcal{A}$ continues to choose a number of adaptive databases and requests subjected to the same conditions, and submits them to $\mathcal{C}$ for responses.

Guess: $\mathcal{A}$ takes a guess b' of b.

The PPT adversary $\mathcal{A}$ can access the data record encryption algorithm METY.*DataEnc* to obtain the plaintext-ciphertext pairs $(\boldsymbol{S}'_j, E(\boldsymbol{S}'_j))$ on jth database encryption. Specifically, $E(\vec{s'}_{j,i|i\in[1,N]})$ is equal to $(\Gamma \cdot \vec{S'}_{j,i} + \vec{e'}_{j,i}) \times \mathcal{M}$, where $\Gamma \in \mathbb{Z}_q$ is a random number, $\vec{e'}_{j,i} \in \mathbb{Z}_q^{2n}$ is a random integer noise vector, $\mathcal{M} \in \mathbb{R}_q^{2n\times 2n}$ is a random invertible matrix, and $\vec{S'}_{j,i} \in \mathbb{Z}_p^{2n}$ is a vector extended by $\vec{s'}_{j,i}$ $(q \gg p)$. Since Γ is the public key, $\mathcal{A}$ can obtain $E(\vec{s'}_{j,i})/\Gamma = \vec{S'}_{j,i} \times \mathcal{M} + (\vec{e'}_{j,i} \times \mathcal{M})/\Gamma$. Under the random perturbation of the random noise vector $\vec{e'}_{j,i} \in \mathbb{Z}_q^{2n}$ and the random matrix $\mathcal{M} \in \mathbb{R}_q^{2n\times 2n}$, it is impossible for $\mathcal{A}$ to recover $\vec{s'}_{j,i}$ from $E(\vec{s'}_{j,i})/\Gamma$. Meanwhile, Independent Component Analysis (ICA) technique, which can successfully attack another matrix encryption scheme named ASPE under ciphertext-only attack [30], also cannot effectively attack METY because $E(\vec{s'}_{j,i})$ contain extra $(\vec{e'}_{j,i} \times \mathcal{M})/\Gamma$. With the extra noise, $\mathcal{A}$ cannot represent the ciphertexts by linearly combining $\{-\frac{1}{2}\sum_{k=1}^n s_{ik}^2, s_{ik} | k \in [1, n]\}$, then it cannot use ICA technique to attack METY. Besides, given the comparison request $\vec{q}_j$ and two data records $\vec{s'}_{j,a}$, $\vec{s'}_{j,b}$, an equation $Comp_a - Comp_b = -\frac{r}{2}(\|\vec{s'}_{j,a} - \vec{q}_j\|^2 - \|\vec{s'}_{j,b} - \vec{q}_j\|^2)$ can be constructed, i.e., $\mathcal{A}$ can launch a linear analysis attack [31]. However, due to the existence of the random number r, the linear analysis attack is not feasible in METY, and the details can refer to [32], [33]. In conclusion, without these unknown elements, $\mathcal{A}$ cannot recover the data record encryption, and cannot distinguish $\vec{s}_{0,i}$ and $\vec{s}_{1,i}$. Considering there are N different ciphertexts in $E(\boldsymbol{S}_0)$ and $E(\boldsymbol{S}_1)$ respectively, the number increases the difficulty of distinguishing $\boldsymbol{S}_0$ and $\boldsymbol{S}_1$. In other words, the probability of $Adv_{\text{METY},\mathcal{A}}^{\text{IND-CLS-CPA-Record}}$ is negligible. □

Definition 4. (IND-CLS-CPA Request Privacy) To capture the IND-CLS-CPA request privacy of METY over security parameter λ, a security game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ is defined as:

Initial: $\mathcal{A}$ sends two scaled request vectors $\vec{q}_0$ and $\vec{q}_1$ of the same dimension to $\mathcal{C}$.

Setup: $\mathcal{C}$ runs METY.*KeyGen*(1^λ) to generate private key sk, and public key pk.

Phase 1: $\mathcal{A}$ submits a number of adaptive chosen databases and requests to $\mathcal{C}$ for encryption:

- Database encryption (METY.*DataEnc*): On the jth database encryption, $\mathcal{A}$ selects a scaled database $\boldsymbol{S}_j = \{\vec{s}_{j,1}, \ldots, \vec{s}_{j,N}\}$, and sends it to $\mathcal{C}$. $\mathcal{C}$ will answer it with encrypted dataset $E(\boldsymbol{S}_j)$ by running METY.*DataEnc*$(\boldsymbol{S}_j, sk, pk)$, where $\boldsymbol{S}_j$ should meet the following conditions simultaneously.
  1) $\mathcal{L}(\boldsymbol{S}_j, \vec{q}_0) = \mathcal{L}(\boldsymbol{S}_j, \vec{q}_1)$;
  2) For $1 \le i \le N$, $\vec{s}_{j,i}$ is included in the top-k records similar to $\vec{q}_0$ and $\vec{q}_1$ in the same order, or it is not in the top-k records, where $1 \le k \le N$.
- Request encryption (METY.*TrapGen*): On the jth request encryption, $\mathcal{A}$ selects a scaled request $\vec{q'}_j$ and sends it to $\mathcal{C}$. $\mathcal{C}$ will answer it with trapdoor $E(\vec{q'}_j)$ by executing METY.*TrapGen*$(\vec{q'}_j, sk, pk)$.

Challenge: With $\vec{q}_0$ and $\vec{q}_1$ received in phase *Initial*, $\mathcal{C}$ throws a coin to decide $b = 0$ or $b = 1$, then returns $E(\vec{q}_b)$ to $\mathcal{A}$ by METY.*TrapGen*$(\vec{q}_b, sk, pk)$.

Phase 2: Same as Phase 1, $\mathcal{A}$ continues to choose a number of adaptive databases and requests subjected to the same conditions, and submit them to $\mathcal{C}$ for responses.

Guess: $\mathcal{A}$ takes a guess b' of b.

Denote the advantage of any probabilistic polynomial time (PPT) adversary $\mathcal{A}$ in guessing $b' = b$ in the game above as $Adv_{\text{METY},\mathcal{A}}^{\text{IND-CLS-CPA-Request}}$, METY is said to be IND-CLS-CPA request privacy *iff* the advantage is negligible.

Theorem 2. METY achieves IND-CLS-CPA request privacy under the above-defined game.

Proof. The trapdoor is generated in a similar way to generating encrypted data records, thus the security analysis of IND-CLS-CPA request privacy can be proved as in *Theorem* 1. For simplicity, here we omit the detailed proof process. □

C. Security Analysis of SMDC

In this subsection, we first give the security definitions of SMDC based on $\mathcal{L}(\boldsymbol{S}, \vec{q})$, and then analyze the security of SMDC. Similar to METY, we still denote the privacy protection of data records and comparison request as *Record Privacy* and *Request Privacy* and strictly follow the definition of IND-CLS-CPA.

Definition 5. (IND-CLS-CPA Record Privacy) To capture the IND-CLS-CPA record privacy of SMDC, a security game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ is defined as:

Initial: $\mathcal{A}$ first selects two scaled databases $\boldsymbol{S}_0 = \{\vec{s}_{0,1}, \vec{s}_{0,2}, \ldots, \vec{s}_{0,N}\}$, $\boldsymbol{S}_1 = \{\vec{s}_{1,1}, \vec{s}_{1,2}, \ldots, \vec{s}_{1,N}\}$ of the same size, then computes and scales the inverse of their covariance matrices $\Sigma_{\boldsymbol{S}_0}^{-1}, \Sigma_{\boldsymbol{S}_1}^{-1}$. Finally, $\mathcal{A}$ sends $\{\boldsymbol{S}_b, \Sigma_{\boldsymbol{S}_b}^{-1}\}_{b\in\{0,1\}}$ to $\mathcal{C}$.

Setup: $\mathcal{C}$ runs SMDC.*KeyGen*(1^λ) to generate private key sk, and public key pk.

Phase 1: $\mathcal{A}$ submits a number of adaptive chosen databases and requests to $\mathcal{C}$ for encryption:

- Database encryption (SMDC.*DataEnc*): On the jth database encryption, $\mathcal{A}$ selects a scaled dataset $\boldsymbol{S}'_j = \{\vec{s'}_{j,1}, \ldots, \vec{s'}_{j,N}\}$, computes and scales its inverse of covariance matrices $\Sigma_{\boldsymbol{S}'_j}^{-1}$, then sends $\{\boldsymbol{S}'_j, \Sigma_{\boldsymbol{S}'_j}^{-1}\}$ to $\mathcal{C}$. $\mathcal{C}$ will answer it with encrypted database $E(\boldsymbol{S}'_j)$ by running SMDC.*DataEnc*$(\boldsymbol{S}'_j, \Sigma_{\boldsymbol{S}'_j}^{-1}, sk, pk)$.
- Request encryption (SMDC.*TrapGen*): On the jth request encryption, $\mathcal{A}$ selects a scaled request $\vec{q}_j$ and sends it to $\mathcal{C}$. $\mathcal{C}$ will answer it with trapdoor $E(\vec{q}_j)$ by running SMDC.*TrapGen*$(\vec{q}_j, sk, pk)$, where $\vec{q}_j$ should meet the following conditions simultaneously.
  1) $\mathcal{L}(\boldsymbol{S}_0, \vec{q}_j) = \mathcal{L}(\boldsymbol{S}_1, \vec{q}_j)$;
  2) For $1 \le i \le N$, $\vec{s}_{0,i}$ and $\vec{s}_{1,i}$ are included in the top-k records similar to $\vec{q}_j$ in the same order, or neither of them are in, where $1 \le k \le N$.

Challenge: With $\{\boldsymbol{S}_0, \Sigma_{\boldsymbol{S}_0}^{-1}\}$ and $\{\boldsymbol{S}_1, \Sigma_{\boldsymbol{S}_1}^{-1}\}$ received in phase *Initial*, $\mathcal{C}$ throws a coin to decide $b = 0$ or $b = 1$, then returns $E(\boldsymbol{S}_b)$ to $\mathcal{A}$ by SMDC.*DataEnc*$(\boldsymbol{S}_b, \Sigma_{\boldsymbol{S}_b}^{-1}, sk, pk)$.

Phase 2: Same as Phase 1, $\mathcal{A}$ continues to choose a number of adaptive databases and requests subjected to the same conditions, and submit them to $\mathcal{C}$ for responses.

Guess: $\mathcal{A}$ takes a guess b' of b.

Denote the advantage of any probabilistic polynomial time (PPT) adversary $\mathcal{A}$ in guessing $b' = b$ in the game above as $Adv_{SMDC,\mathcal{A}}^{\text{IND-CLS-CPA-Record}}$, SMDC is said to be IND-CLS-CPA record privacy *iff* the advantage is negligible.

Theorem 3. SMDC achieves IND-CLS-CPA record privacy under the above-defined game.

Proof. The security game defined in *Definition* 5 is exploited to analyze the security of our designed comparison method SMDC.

Initial: $\mathcal{A}$ first selects two scaled databases $\boldsymbol{S}_0 = \{\vec{s}_{0,1}, \vec{s}_{0,2}, \ldots, \vec{s}_{0,N}\}$, $\boldsymbol{S}_1 = \{\vec{s}_{1,1}, \vec{s}_{1,2}, \ldots, \vec{s}_{1,N}\}$ of the same size, then computes and scales the inverse of their covariance matrices $\Sigma_{\boldsymbol{S}_0}^{-1}, \Sigma_{\boldsymbol{S}_1}^{-1}$. Finally, $\mathcal{A}$ sends $\{\boldsymbol{S}_b, \Sigma_{\boldsymbol{S}_b}^{-1}\}_{b\in\{0,1\}}$ to $\mathcal{C}$, note that the dimension of each $\vec{s}_{b,i}$ is n.

Setup: $\mathcal{C}$ runs SMDC.*KeyGen*(1^λ) to generate private key $sk = \mathcal{M}$, and public key $pk = \Gamma$.

Phase 1: $\mathcal{A}$ submits a number of adaptive chosen databases and requests to $\mathcal{C}$ for encryption:

- Database encryption (SMDC.*DataEnc*): On the jth database encryption, $\mathcal{A}$ first selects a scaled database $\boldsymbol{S}'_j = \{\vec{s'}_{j,1}, \ldots, \vec{s'}_{j,N}\}$, then computes and scales its inverse of covariance matrices $\Sigma_{\boldsymbol{S}'_j}^{-1}$. Finally, it sends $\{\boldsymbol{S}'_j, \Sigma_{\boldsymbol{S}'_j}^{-1}\}$ to $\mathcal{C}$. $\mathcal{C}$ will answer it with encrypted database $E(\boldsymbol{S}'_j) = \{E(\vec{s'}_{j,i})|_{i=1}^N\}$ by SMDC.*DataEnc*$(\boldsymbol{S}'_j, \Sigma_{\boldsymbol{S}'_j}^{-1}, sk, pk)$, where each record $\vec{s'}_{j,i}$ in $\boldsymbol{S}'_j$ is encrypted to $E(\vec{s'}_{j,i}) = \mathcal{M}^{-1} \times (\Gamma \cdot \vec{S'}_{j,i} + \vec{e'}_{j,i}^{\,T})$, and each $\vec{S'}_{j,i} = (\underbrace{-2 \cdot \Sigma_{\boldsymbol{S}'_j}^{-1} \vec{s'}_{j,i}^{\,T}}_{n}, \underbrace{\vec{s'}_{j,i} \Sigma_{\boldsymbol{S}'_j}^{-1} \vec{s'}_{j,i}^{\,T}}_{1}, \alpha)^T$, $\vec{e'}_{j,i}^{\,T}$ is a random integer noise vector.
- Request encryption (SMDC.*TrapGen*): On the jth request encryption, $\mathcal{A}$ selects a scaled n-dimensional request vector $\vec{q}_j$ and sends it to $\mathcal{C}$. $\mathcal{C}$ answers it with trapdoor $E(\vec{q}_j) = (\Gamma \cdot \vec{Q} + \vec{e}_q) \times \mathcal{M}$ by running SMDC.*TrapGen*$(\vec{q}_j, sk, pk)$, where $\vec{Q} = (\underbrace{r \cdot \vec{q}_j}_{n}, r, \beta)$, and $\vec{e}_q$ is a random integer noise vector. Besides, $\vec{q}_j$ also should meet the following conditions simultaneously.
  1) $\mathcal{L}(\boldsymbol{S}_0, \vec{q}_j) = \mathcal{L}(\boldsymbol{S}_1, \vec{q}_j)$;
  2) For $1 \le i \le N$, $\vec{s}_{0,i}$ and $\vec{s}_{1,i}$ are included in the top-k records similar to $\vec{q}_j$ in the same order, or neither of them are in, where $1 \le k \le N$.

Challenge: With $\{\boldsymbol{S}_0, \Sigma_{\boldsymbol{S}_0}^{-1}\}$ and $\{\boldsymbol{S}_1, \Sigma_{\boldsymbol{S}_1}^{-1}\}$ received in phase *Initial*, $\mathcal{C}$ throws a coin to decide $b = 0$ or $b = 1$, then returns $E(\boldsymbol{S}_b)$ to $\mathcal{A}$ by SMDC.*DataEnc*$(\boldsymbol{S}_b, \Sigma_{\boldsymbol{S}_b}^{-1}, sk, pk)$.

Phase 2: Same as Phase 1, $\mathcal{A}$ continues to choose a number of adaptive databases and requests subjected to the same conditions, and submits them to $\mathcal{C}$ for responses.

Guess: $\mathcal{A}$ takes a guess b' of b.

The PPT adversary $\mathcal{A}$ can access the data record encryption algorithm SMDC.*DataEnc* to obtain the plaintext-ciphertext pairs $(\boldsymbol{S}'_j, E(\boldsymbol{S}'_j))$ on jth database encryption. However, the adversary does not know the private key $\mathcal{M}$, the random number α and the random integer noise vectors $\vec{e'}_{j,i}, 1 \le i \le N$. Therefore, according to each pair $(\vec{s'}_{j,i}, E(\vec{s'}_{j,i}))$ in $(\boldsymbol{S}'_j, E(\boldsymbol{S}'_j))$, $\mathcal{A}$ can only construct $(n+2)$ dot products of $(n+2)$-dimensional vectors like [20]. Meanwhile, the encrypted record $E(\vec{s'}_{j,i}) = \mathcal{M}^{-1} \times (\Gamma \cdot \vec{S'}_{j,i} + \vec{e'}_{j,i}^{\,T})$ can be represented as $\tilde{\mathcal{M}} \times \vec{S'}_{j,i} + \tilde{e'}_{j,i}^{\,T}$, where $\tilde{\mathcal{M}} = \Gamma \cdot \mathcal{M}^{-1}$ is a random matrix, $\vec{S'}_{j,i}$ is a vector included a random number α and $\tilde{e'}_{j,i}^{\,T}$ is a random integer noise vector. According to the security analysis of METY, we can know that recovering $\vec{s'}_{j,i}$ from $E(\vec{s'}_{j,i})$ through signal processing and linear analysis attack is impossible. If it is impossible to recover the owner's records directly from their ciphertexts, recovering $\mathcal{M}$, α and $\vec{e'}_{j,i}$ is also difficult for $\mathcal{A}$. Without these unknown elements, $\mathcal{A}$ cannot distinguish $\vec{s}_{0,i}$ and $\vec{s}_{1,i}$, and N different ciphertexts respectively in $E(\boldsymbol{S}_0)$ and $E(\boldsymbol{S}_1)$ increases the difficulty of distinguishing $\boldsymbol{S}_0$ and $\boldsymbol{S}_1$. As a result, the probability of $Adv_{SMDC,\mathcal{A}}^{\text{IND-CLS-CPA-Record}}$ is negligible. □

Definition 6. (IND-CLS-CPA Request Privacy) To capture the IND-CLS-CPA request privacy of SMDC over security parameter λ, a security game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ can be defined like the game in *Definition 4*. To avoid repetition, the details are omitted here.

Denote the advantage of any probabilistic polynomial time (PPT) adversary $\mathcal{A}$ in guessing the coin in the game above as $Adv^{\text{IND-CLS-CPA-Request}}_{\text{SMDC},\mathcal{A}}$, SMDC is said to be IND-CLS-CPA request privacy *iff* the advantage is negligible.

Theorem 4. SMDC achieves IND-CLS-CPA request privacy under the above-defined game.

Proof. Since the trapdoor is generated in a similar way to generating encrypted data records, thus the security analysis of IND-CLS-CPA request privacy can refer to the proof of *Theorem* 3. □

D. Security Analysis of NAIAD

In NAIAD, the confidentiality of pre-diagnostic results can be achieved through PKE algorithm and the disease index table DT. By PKE, the same labels associated with different medical records can be encrypted into different ciphertexts, thus without sk_E and DT, no one can read the returned labels and the final pre-diagnostic result. Therefore, we focus on the privacy of $\mathcal{T}$'s indexes and query requests, as well as the trapdoor unlinkability in this subsection.

Theorem 5. NAIAD is secure under IND-CLS-CPA, *iff* SMDC is secure under IND-CLS-CPA.

Proof. We prove the security of NAIAD by proving that NAIAD can achieve IND-CLS-CPA index privacy and request privacy.

Index privacy. In NAIAD, a hierarchical index tree $\mathcal{T}$ is constructed over the medical database, and the indexes include non-leaf nodes $\{\Sigma^{-1}_{h,d}, \vec{c}_{h,d}\}(h > 0)$ and leaf nodes $\{C_{0,d}, \Sigma^{-1}_{0,d}, \vec{c}_{0,d}\}$. Obviously, the indexes consist of clusters $C_{0,d}$ and center points with the inverse of covariance matrices $\{\Sigma^{-1}_{h,d}, \vec{c}_{h,d}\}(h \geq 0)$. Since each cluster $C_{0,d}$ is encrypted by SMDC.*DataEnc* and the record privacy of SMDC has been proven in *Theorem* 3, the security of SMDC can prevent an adversary from obtaining any medical records stored in the clusters. Moreover, each center point with covariance matrix $\{\Sigma^{-1}_{h,d}, \vec{c}_{h,d}\}(h \geq 0)$ is encrypted by NAIAD.*NodeEnc*. Considering that the used private keys $\mathcal{M}_{\omega|1\leq\omega\leq n}$, $\delta_{\omega|1\leq\omega\leq n}$, random numbers $\mu_{\omega 1}, \mu_{\omega 2}, \alpha'$, and the random integer noise vectors $\vec{e}_{h,d}$, $\vec{e}_{\omega|1\leq\omega\leq n}$ make recovering $\{\Sigma^{-1}_{h,d}, \vec{c}_{h,d}\}$ from ciphertexts into difficult linear equation solving problems, the essence of this encryption algorithm NAIAD.*NodeEnc* is the same as SMDC.*DataEnc*. Thus, when SMDC achieves IND-CLS-CPA record privacy, NAIAD can also achieve IND-CLS-CPA index privacy.

Request privacy. In NAIAD, a trapdoor (i.e., encrypted query request) consists of $E(\vec{q})$ and $E(\vec{\gamma}_\omega)|^n_{\omega=1}$. Specifically, $E(\vec{q})$ is achieved by SMDC.*TrapGen*, thus the security of SMDC can guarantee the security of $E(\vec{q})$. And $E(\vec{\gamma}_{\omega|1\leq\omega\leq n})$ is obtained by encrypting $q_\omega \cdot \vec{q}$, where q_ω is the ωth element in $\vec{q}$. To prevent this operation leaks the content of request vector, n integers $\delta_{\omega|1\leq\omega\leq n}$ are introduced to randomized $q_\omega \cdot \vec{q}$, then they are encrypted by the same way as SMDC.*TrapGen*. Thus, when SMDC achieves IND-CLS-CPA request privacy, NAIAD can also achieve IND-CLS-CPA request privacy.

As mentioned above, the security of NAIAD can be proven.□

Theorem 6. NAIAD achieves trapdoor unlinkability among different encrypted query requests.

Proof. When a legal patient wants to submit a query request $\vec{q}$, it first needs to be encrypted into $\{E(\vec{\gamma}_\omega)|^n_{\omega=1}, E(\vec{q})\}$:

$$E(\vec{\gamma}_\omega) = (\Gamma \cdot \vec{\Upsilon}_\omega + \vec{e}^*_\omega) \times \mathcal{M}_\omega,$$

$$E(\vec{q}) = (\Gamma \cdot \vec{Q} + \vec{e}_q) \times \mathcal{M}_0.$$

Since the $(n+1)$ unknown random noise vector $\vec{e}^*_{\omega|1\leq\omega\leq n}$ and $\vec{e}_q$ selected by each request are different, the trapdoor generated each time is different even for the same query request. Moreover, when extending $\vec{q}$ to $\vec{\Upsilon}_{\omega|1\leq\omega\leq n}$ and $\vec{Q}$, a total of $(2n+1)$ random numbers are introduced, which further ensures that the generated trapdoors from the same query request will not be same. Therefore, it is computationally infeasible for a cloud server to determine whether two trapdoors are generated from the same request. That means, NAIAD can achieve trapdoor unlinkability among different encrypted query requests. □

VI. Performance Evaluation

In this section, we demonstrate the performance of SMDC and NAIAD from two perspectives of theoretical analysis and experimental evaluations, and make some comparisons with the popular pre-diagnostic models and the state-of-the-art similar work in terms of accuracy and efficiency.

A. Theoretical Analysis

The complexity comparison of SMDC, NAIAD with PMDC, TAMMIE proposed in [15] is given in Table II. Specifically, SMDC and PMDC achieve privacy-preserving Mahalanobis distance comparison under the same covariance matrix; while NAIAD and TAMMIE securely compare Mahalanobis distances under different covariance matrices. The comparison consists of three main phases: *DataEnc*, *TrapGen* and *DisComp*. In addition, the assumption and nations used in this subsection are given as follows: there is only one non-revoked HI and PT; N is the size of the database; n is the dimension of the data, $n_o = n + 1$ and $n_t = n + 2$; K is the number of clusters; N_C is the number of data stored in the chosen cluster; $N_\mathcal{T}$ is the number of nodes consist of a hierarchical index tree $\mathcal{T}$; and N_R is the number of nodes retrieved on $\mathcal{T}$. For expression simplicity, we will ignore the computational overhead of the calculations that need to be performed in all four schemes, e.g, sorting. We conduct the theoretical analysis from computational complexity and communication complexity.

Computational complexity. Denoting the computational complexity of n-dimensional vector inner product as $\mathcal{O}(n)$, then $\mathcal{O}(n^2)$ can represent the multiplication computational complexity of an $n \times n$-dimensional matrix and an n-dimensional vector, $\mathcal{O}(n^3)$ can represent the multiplication computational complexity of two $n \times n$-dimensional matrices. In PMDC, each vector is expanded to n_t dimensions and encrypted into two

TABLE II
COMPLEXITY COMPARISON OF VARIOUS SCHEMES

Schemes	Computational complexity			Communication complexity		
	DataEnc	**TrapGen**	**DisComp**	**DataEnc**	**TrapGen**	**DisComp**
PMDC [15]	$\mathcal{O}(6N \cdot n_t^3)$	$\mathcal{O}(6 \cdot n_t^3)$	$\mathcal{O}(2N \cdot n_t^2)$	$\mathcal{O}(2N \cdot n_t^2)$	$\mathcal{O}(2 \cdot n_t^2)$	$\mathcal{O}(N)$
SMDC	$\mathcal{O}(N \cdot n_t^2)$	$\mathcal{O}(n_t^2)$	$\mathcal{O}(N \cdot n_t)$	$\mathcal{O}(N \cdot n_t)$	$\mathcal{O}(n_t)$	$\mathcal{O}(N)$
TAMMIE [15]	$\mathcal{O}((6N + 9K)n_t^3)$	$\mathcal{O}((3K + 6)n_t^3)$	$\mathcal{O}((3K + 2N_C)n_t^2)$	$\mathcal{O}((2N + 3K)n_t^2)$	$\mathcal{O}((K + 2)n_t^2)$	$\mathcal{O}(k)$
NAIAD	$\mathcal{O}([N + n_o N_{\mathcal{T}}]n_t^2)$	$\mathcal{O}(n_o n_t^2)$	$\mathcal{O}([n_o N_R + N_C]n_t)$	$\mathcal{O}([N + n_o N_{\mathcal{T}}]n_t)$	$\mathcal{O}(n_o n_t)$	$\mathcal{O}(k)$

parts by a total of six matrix multiplications, thus the computational complexity of *DataEnc* and *TrapGen* are $\mathcal{O}(6N \cdot n_t^3)$ and $\mathcal{O}(6 \cdot n_t^3)$ respectively. In *Discomp*, the trace of a matrix obtained by multiplying two $n_t \times n_t$-dimensional matrices needs to be computed $2\tilde{}N$ times, and computing such a trace is equivalent to computing the inner product of two n_t-dimensional vectors n_t times, thus the computational complexity of *Discomp* is $\mathcal{O}(2N \cdot n_t^2)$. Different from PMDC, the encryption of vectors in SMDC is realized by the multiplication of an n_t-dimensional vector and an $n_t \times n_t$-dimensional matrix, the distance comparisons are realized by computing the inner product of two n_t-dimensional vectors N times. Therefore, the computational complexity of *DataEnc*, *TrapGen* and *Discomp* in SMDC are $\mathcal{O}(N \cdot n_t^2)$, $\mathcal{O}(n_t^2)$ and $\mathcal{O}(N \cdot n_t)$ in respective. Compared with PMDC, TAMMIE also clusters the similar data into K clusters to improve the query efficiency, the covariance matrices of the K clusters are different. To compare the Mahalanobis distances over ciphertexts under different covariance matrices, apart from the basic encryption operations, TAMMIE performs extra $9\tilde{}K$ matrix multiplications for covariance matrices and extra $3\tilde{}K$ matrix multiplications for a request vector. Besides, the distance comparison over the cluster centers also requires $3\tilde{}K$ calculations for matrix trace. Therefore, the computational complexity of *DataEnc*, *TrapGen* and *Discomp* in TAMMIE are $\mathcal{O}((6N + 9\tilde{}K)n_t^3)$, $\mathcal{O}((3K + 6)n_t^3)$ and $\mathcal{O}((3K + 2N_C)n_t^2)$ respectively. In NAIAD, a hierarchical index tree $\mathcal{T}$ is constructed over the K clusters, except for the vectors stored in the dataset, $\mathcal{T}$ also contains $N_{\mathcal{T}}$ center points, to encrypt these center points, $N_{\mathcal{T}} \cdot n_o$ vector-matrix multiplications need to be performed. And the request is encrypted n_o times as in SMDC. When retrieving $\mathcal{T}$, the computation with each retrieved node requires extra n_o inner products compared with *Discomp* in SMDC. Therefore, the computational complexity of *DataEnc*, *TrapGen* and *Discomp* in NAIAD are $\mathcal{O}([N + n_o N_{\mathcal{T}}]n_t^2)$, $\mathcal{O}(n_o n_t^2)$ and $\mathcal{O}([n_o N_R + N_C]n_t)$ in respective.

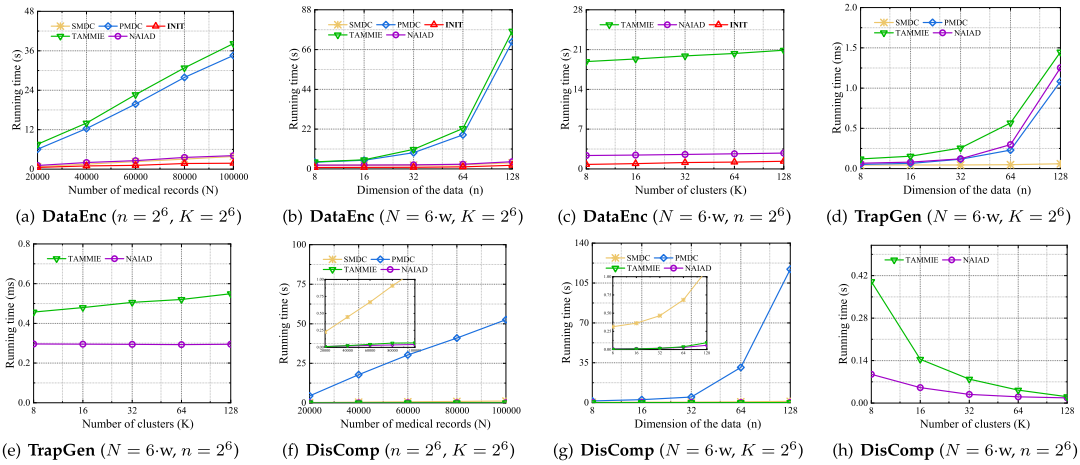
Communication complexity. Assuming that the communication complexity of an n-dimensional vector is $\mathcal{O}(n)$, and the communication complexity of an $n \times n$-dimensional matrix is $\mathcal{O}(n^2)$. In PMDC and SMDC, each original vector is encrypted into two $n_t \times n_t$-dimensional matrices and one n_t-dimensional vector respectively, and the result of distance comparison is returned, thus the communication complexity of *DataEnc*, *TrapGen*, *Discomp* in PMDC and SMDC are $\mathcal{O}(2\tilde{}Nn_t^2)$, $\mathcal{O}(2n_t^2)$, $\mathcal{O}(N)$ and $\mathcal{O}(Nn_t)$, $\mathcal{O}(n_t)$, $\mathcal{O}(N)$ respectively. Due to the introduction of clusters and tree, TAMMIE and NAIAD require additional communication overhead. Specifically, in TAMMIE, the cluster centers are encrypted into three parts, each part is an $n_t \times n_t$-dimensional matrix, and the request is encrypted into $K + 2$ matrices to calculate with different cluster centers. Therefore, the communication complexity of *DataEnc* and *TrapGen* are $\mathcal{O}((2N + 3\tilde{}K)n_t^2)$ and $\mathcal{O}((K + 2)n_t^2)$ in respective. In NAIAD, the cluster centers in $\mathcal{T}$ and request are all encrypted into n_o n_t-dimensional vectors, thus the communication complexity of *DataEnc* and *TrapGen* are $\mathcal{O}([N + n_o N_{\mathcal{T}}]n_t)$ and $\mathcal{O}(n_o n_t)$ in respective. Since TAMMIE and NAIAD will return top-k similar vectors to a query request, their communication complexity of *DisComp* are both $\mathcal{O}(k)$.

B. Experimental Evaluations

In order to measure the integrated performance, we implement SMDC and NAIAD with Python on a computer with 2.3 GHz Intel i5 four-core processor, 8 GB memory, and macOS High Sierra system. Specifically, the parameters used in our proposed scheme are set as $\rho = 1000$, $|p| = 20$bits, $|q| = 30$bits, n varies from 8 to 128 in the tests, where $|\cdot|$ denotes the bit length, $|q| = |p| + 10$ can ensure the correctness and security (the security can be further improved by increasing the bit length and extending n by padding random numbers). The evaluations for accuracy are conducted on five real-world datasets chosen from UCI machine learning repository [34], and the evaluations for efficiency are conducted on a randomly generated synthetic dataset. The datasets used are described in detail below.

- Real-world datasets. 1) *HCV* consists of 615 HCV medical records with 13 attributes and 5 diagnoses; 2) *Breast* consists of 569 breast cancer medical records with 30 attributes and 2 diagnoses; 3) *HEART* consists of 303 heart disease medical records with 13 attributes and 2 diagnoses; 4) *HEARTF* consists of 299 heart failure medical records with 12 attributes and 2 diagnoses; 5) *OBESITY* consists of 2111 obesity medical records with 16 attributes and 6 diagnoses.
- Synthetic dataset. The synthetic dataset is randomly generated and contains 100000 vectors with different dimensions ranging from 8 to 128. It mainly used to test how different factors affect the efficiency of SMDC, NAIAD, PMDC and TAMMIE.

1) Accuracy: To test the accuracy of our proposed schemes, we randomly divide each of the five real-world datasets into two partitions, 70% is the training set and 30% is the testing set. SMDC and NAIAD are constructed using the same similarity metric, Mahalanobis distance (MD), and SMDC can be regarded as a scheme for finding top-k similar records on one cluster. Therefore, we denote the pre-diagnostic model used in SMDC and NAIAD as MD-based. Specifically, MD-based model first



(a) **DataEnc** ($n = 2^6$, $K = 2^6$) (b) **DataEnc** ($N = 6{\cdot}$w, $K = 2^6$) (c) **DataEnc** ($N = 6{\cdot}$w, $n = 2^6$) (d) **TrapGen** ($N = 6{\cdot}$w, $K = 2^6$)

(e) **TrapGen** ($N = 6{\cdot}$w, $n = 2^6$) (f) **DisComp** ($n = 2^6$, $K = 2^6$) (g) **DisComp** ($N = 6{\cdot}$w, $K = 2^6$) (h) **DisComp** ($N = 6{\cdot}$w, $n = 2^6$)

Fig. 4. Average computational cost of SMDC, NAIAD, PMDC and TAMMIE (test 1000 times), where w $= 10^4$.

TABLE III
ACCURACY OF DIFFERENT PRE-DIAGNOSTIC MODEL

Classifier	HCV	Breast	HEART	HEARTF	OBESITY
MD-based	**0.972**	0.912	**0.813**	**0.744**	**0.836**
NB	0.893	0.906	0.714	0.667	0.543
kNN-3	0.904	**0.918**	0.571	0.622	**0.815**
kNN-7	0.898	0.906	0.593	0.644	0.811
SVM-L	**0.955**	**0.953**	**0.802**	**0.811**	0.812
SVM-P	0.921	0.906	0.626	0.644	0.591

The bold entities indicate the two most accurate data in a column.

groups the similar medical records into a cluster, then calculates each cluster's covariance matrix and center point, the diagnosis is determined by the labels of the top-k similar medical records in the closest cluster. Furthermore, with the same training and testing sets, we also evaluate the accuracy of some popular machine learning classifiers in medical pre-diagnosis, namely, Naïve Bayesian (NB), euclidean distance based k-Nearest Neighbors (kNN) and Support Vector Machine (SVM).

Table III shows the accuracy of different pre-diagnostic models over different medical datasets, it can be observed that the MD-based model used in our proposed schemes consistently performs well. Although linear SVM performs better in *Breast* and *HEARTF*, its training process is more complicated than that of MD-based. Consequently, SMDC and NAIAD can achieve a high accuracy pre-diagnosis with a simple model.

2) Efficiency: The synthetic dataset is used in this subsection to conduct comparison experiments with PMDC and TAMMIE. Specifically, the comparison consists of three phases: *DataEnc*, *TrapGen* and *Diacomp*. And compared with TAMMIE, NAIAD takes extra time to construct $\mathcal{T}$, this is one-time consuming, we denote it as *INIT* and test it in *DataEnc* phase. Moreover, learning from the analysis in Section VI-A, the computational cost of SMDC and PMDC is mainly affected by the size of the database N and the dimension of the data n, while the computational cost of NAIAD and TAMMIE is also affected by the number of clusters K.

DataEnc. Taking different influencing factors (i.e., N, n and K) into consideration, Fig. 4(a), (b), and (c) respectively describe the average running time of *DataEnc* varying with them. Both n and K are set to 64, when N ranges from 20000 to 100000, the running time of *DataEnc* in SMDC, PMDC, NAIAD and TAMMIE all increases linearly as shown in Fig. 4(a). Given $N = 60000$, $K = 64$, Fig. 4(b) is depicted with n from 8 to 128. As n increases, it is easy to observe that the running time in all schemes increases. In Fig. 4(c), we set $N = 60000$, $n = 64$, varying K from 8 to 128, we can see that the larger K is, the longer the running time of NAIAD and TAMMIE will be. This is because more nodes and clusters appearing in $\mathcal{T}$ require more encryption operations. From Fig. 4(a), (b), and (c), we can observe that the maximum running time of SMDC is 3.8511 s while that of PMDC is 70.3047 s and the maximum running time of NAIAD is 4.1113 s while that of TAMMIE is 76.2015 s. It is obvious that SMDC performs better than PMDC and NAIAD performs better than TAMMIE. Although *INIT* brings extra overhead to NAIAD, and the overhead increases varying with the increase of N, n and K, it only runs one-time and costs only up to 2.0071 s.

TrapGen. Since the computational cost of *TrapGen* is only affected by n and K, we draw Fig. 4(d) and Fig. 4(e) with $N = 60000$. Given $K = 64$, Fig. 4(d) is depicted with n from 8 to 128, due to the encryption matrix becomes larger with the increase of n, the running time in all schemes grows when n grows. When n is set to 64, K ranges from 8 to 128, Fig. 4(e) shows that NAIAD is not influenced, but the speed of *TrapGen* in TAMMIE is reduced with the increase of K. In this phase, the running time of SMDC and NAIAD is at most 0.0603 ms and 1.2519 ms respectively, which is still faster than PMDC and TAMMIE (up to 1.0820 ms and 1.4477 ms, respectively), and is only affected by n.

Diacomp. Figs. 4(f), (g), and (h) depict the computational cost of *DisComp* varying with N, n and K respectively. In Fig. 4(f), n and K are set to 64, N ranges from 20000 to 100000, the figure shows that the running time of *DisComp* linearly grows with the increase of N since more similarity calculations are required for finding top-k medical records. When N is set to 60000, K is set to 64, Fig. 4(g) draws the running time of all schemes grows with n ranges from 8 to 128. This is because higher dimensions result in higher complexity matrix multiplications. In Fig. 4(h), we set $N = 60000$, $n = 64$, varying K from 8 to

128, we can see that the running time of NAIAD and TAMMIE decreases with the increase of K because of the narrowing of the retrieval range. From Fig. 4(g) and (h), we can see that the maximum running time of SMDC and PMDC are 1.1268 s and 117.0700 s respectively, it is obvious that SMDC performs better in the basic comparison methods. Meanwhile, the maximum running time of NAIAD and TAMMIE are 0.0579 s and 0.0941 s respectively, which shows the introduction of the hierarchical index tree further improves the retrieval efficiency. Therefore, when the medical pre-diagnosis scheme is applied in practice, NAIAD has better advantages in terms of efficiency.

VII. Related Work

In this section, we briefly review the related work from the perspective of privacy-preserving medical pre-diagnosis and applications of Mahalanobis distance classification.

Privacy-preserving medical pre-diagnosis. With the development of cloud computing, the medical pre-diagnosis services tend to be provided by one or more cloud servers, and the protection of sensitive data has been taken into consideration. Specifically, based on the Paillier cryptosystem [35], Bost et al. designed three secure classification protocols to achieve privacy-preserving disease prediction over several real medical datasets [36]. Similarly, benefiting from the additive homomorphism property, Liu et al. proposed a privacy-preserving patient-centric clinical decision support system based on Naïve Bayesian classification, which can help clinician complementary to diagnose the risk of patients' disease in a secure way [37]; and Hua et al. achieved precise and privacy-preserving pre-diagnosis services by outsourcing an encrypted skyline [38] pre-diagnostic model [10]. The security of all the above schemes are guaranteed by homomorphic encryption. By introducing random masking and polynomial aggregation techniques [39] into online medical services, Zhu et al. proposed two efficient and privacy-preserving pre-diagnosis schemes based on non-linear SVM and ML-kNN [7], [11]. Meanwhile, Zhang et al. utilized random vectors and matrices to enable the encrypted medical records can be handled and trained directly on the cloud server via SLP [40] algorithm [41]. Wang et al. also employed matrix encryption to the Naïve Bayesian classifier and decision tree classifier [9], which can provide online pre-diagnosis by searching similar records without leaking original data. Besides, Ma et al. proposed a lightweight privacy-preserving medical diagnosis scheme in edge computing called LPME, which can reduce transmission latency and provide real-time services over ciphertexts [42]. Sun et al. applied ELGamal Digital Signature to realize secure search of past case-database, which greatly increases the timeliness of information acquisition and meets high-speed information sharing requirements [43]. Recently, taking the privacy of data structure into consideration, Zhang et al. proposed a privacy-preserving decision tree evaluation scheme for medical diagnosis by utilizing the elementary matrix, monotonically increasing and one-way function [44]. The core of privacy-preserving medical pre-diagnosis is to compute similarity securely, and the above-mentioned schemes focus on simple dimension independent similarity metrics. Considering the relationship between the feature of each dimension, Zhang et al. presented a privacy-preserving disease diagnosis scheme based on dual cloud model and homomorphic encryption techniques [45], [46], and they adopted Mahalanobis distance evaluation model that shows better accuracy. However, they do not consider the model privacy as the parameters are owned by cloud servers [14].

Applications of Mahalanobis distance classification. Mahalanobis distance takes into account unequal variances and correlations between features, and excels at classifying dimension related datasets. Xiang et al. applied Mahalanobis distance to data clustering, interactive natural image segmentation and face pose estimation, their proposed scheme achieved significant progress compared to using euclidean distance [47]. Roth et al. demonstrated that Mahalanobis metric learning can yield quite good classification results in the context of single-shot person re-identification [48]. In the medical field, Wei et al. presented a two-step content-based image retrieval scheme for computer-aided diagnosis of lung nodules, and Mahalanobis distance is used to preserve the semantic relevance of extracted vectors [49]. Moreover, Sarmadi et al. proposed a novel ensemble learning-based method for structural health monitoring three kinds of Mahalanobis distance metrics. The proposed method highly succeeds in detecting damage [50]. Furthermore, By combining automatic adaptive feature extraction with Mahalanobis distance classification criterion, Sun et al. presented a novel system for more accurate diagnosis of heart diseases [51]. All of the above Mahalanobis distance based schemes are designed over plaintext. To avoid leaking sensitive information, recently some privacy-preserving Mahalanobis distance comparsion schemes [14], [52], [53] also have been proposed by introducing different homomorphic encryption algorithms (e.g., labeled-homomorphic encryption [54]).

VIII. Conclusion

In this article, we have proposed a privacy-preserving cloud-assisted medical pre-diagnosis scheme named NAIAD. NAIAD can provide patients with high-accurate pre-diagnosis services while avoiding complex machine learning classifiers and heavy encryption calculations. Moreover, the designed secure Mahalanobis-distances similarity comparison methods under the same covariance matrix and different covariance matrices are generic. They can be easily adapted to other secure similarity comparison scenarios (e.g., image retrieval, fingerprint recognition) to improve the accuracy. For the future work, under the premise of maintaining efficiency and precision, we will investigate the dual cloud model, which can support lightweight and robust encryption techniques against stronger attacks, such as chosen-plaintext attack.

References

[1] M. S. Hossain and G. Muhammad, "Cloud-assisted industrial Internet of Things (IIoT)–enabled framework for health monitoring," *Comput. Netw.*, vol. 101, pp. 192–202, 2016.

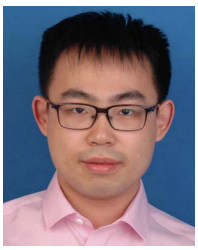
[2] P. M. Kumar, S. Lokesh, R. Varatharajan, G. C. Babu, and P. Parthasarathy, "Cloud and iot based disease prediction and diagnosis system for healthcare using fuzzy neural classifier," *Future Gener. Comput. Syst.*, vol. 86, pp. 527–534, 2018.

[3] J. Liang, Z. Qin, S. Xiao, L. Ou, and X. Lin, "Efficient and secure decision tree classification for cloud-assisted online diagnosis services," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 4, pp. 1632–1644, Jul./Aug. 2021.
[4] C. Dong et al., "Maliciously secure and efficient large-scale genome-wide association study with multi-party computation," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 1243–1257, Mar./Apr. 2023.
[5] J. McKeon, "Houston health department suffers healthcare data breach," Accessed: Mar. 04, 2022. [Online]. Available: https://healthitsecurity.com/news/houston-health-department-suffers-healthcare-data-breach
[6] M. Barua, X. Liang, R. Lu, and X. Shen, "ESPAC: Enabling security and patient-centric access control for ehealth in cloud computing," *Int. J. Secur. Netw.*, vol. 6, no. 2/3, pp. 67–76, 2011.
[7] H. Zhu, X. Liu, R. Lu, and H. Li, "Efficient and privacy-preserving online medical prediagnosis framework using nonlinear SVM," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 3, pp. 838–850, May 2017.
[8] J. Park and D. H. Lee, "Privacy preserving K-nearest neighbor for medical diagnosis in E-health cloud," *J. Healthcare Eng.*, vol. 2018, pp. 1–11, 2018.
[9] X. Wang, J. Ma, M. Yinbin, X. Liu, and Y. Ruikang, "Privacy-preserving diverse keyword search and online pre-diagnosis in cloud computing," *IEEE Trans. Serv. Comput.*, vol. 15, no. 2, pp. 710–723, Mar./Apr. 2022.
[10] J. Hua, G. Shi, H. Zhu, F. Wang, X. Liu, and H. Li, "Camps: Efficient and privacy-preserving medical primary diagnosis over outsourced cloud," *Inf. Sci.*, vol. 527, pp. 560–575, 2020.
[11] D. Zhu et al., "CREDO: Efficient and privacy-preserving multi-level medical pre-diagnosis based on ML-KNN," *Inf. Sci.*, vol. 514, pp. 244–262, 2020.
[12] B. Xie, T. Xiang, X. Liao, and J. Wu, "Achieving privacy-preserving online diagnosis with outsourced SVM in internet of medical things environment," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 6, pp. 4113–4126, Nov./Dec. 2022.
[13] S. Zhang, S. Ray, R. Lu, Y. Zheng, Y. Guan, and J. Shao, "Achieving efficient and privacy-preserving dynamic skyline query in online medical diagnosis," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9973–9986, Jun. 2022.
[14] M. Zhang, Y. Zhang, and G. Shen, "PPDDS: A privacy-preserving disease diagnosis scheme based on the secure mahalanobis distance evaluation model," *IEEE Syst. J.*, vol. 16, no. 3, pp. 4552–4562, Sep. 2022.
[15] D. Zhu, H. Zhu, X. Wang, R. Lu, and D. Feng, "An accurate and privacy-preserving retrieval scheme over outsourced medical images," *IEEE Trans. Serv. Comput.*, to be published, doi: 10.1109/TSC.2022.3149847.
[16] L. Liu et al., "Toward highly secure yet efficient KNN classification scheme on outsourced cloud data," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9841–9852, Dec. 2019.
[17] C. Xu, N. Wang, L. Zhu, C. Zhang, K. Sharif, and H. Wu, "Reliable and privacy-preserving top-k disease matching schemes for E-healthcare systems," *IEEE Internet Things J.*, vol. 9, no. 7, pp. 5537–5547, Apr. 2022.
[18] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 222–233, Jan. 2014.
[19] G. J. McLachlan, "Mahalanobis distance," *Resonance*, vol. 4, no. 6, pp. 20–26, 1999.
[20] J. Yuan and Y. Tian, "Practical privacy-preserving mapreduce based K-means clustering over large-scale dataset," *IEEE Trans. Cloud Comput.*, vol. 7, no. 2, pp. 568–579, Second Quarter 2019.
[21] Z. Brakerski, C. Gentry, and S. Halevi, "Packed ciphertexts in LWE-based homomorphic encryption," in *Proc. Int. Workshop Public Key Cryptogr.*, 2013, pp. 1–13.
[22] 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.
[23] D. Charalampidis, "A modified k-means algorithm for circular invariant clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1856–1865, Dec. 2005.
[24] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in *ACM Conf. Comput. Commun. Secur.*, 2012, pp. 965–976.
[25] X. Wang, J. Ma, X. Liu, Y. Miao, Y. Liu, and R. H. Deng, "Forward/backward and content private DSSE for spatial keyword queries," *IEEE Trans. Dependable Secure Comput.*, to be published, doi: 10.1109/TDSC.2022.3205670.
[26] C. Huang, D. Liu, A. Yang, R. Lu, and X. Shen, "Multi-client secure and efficient DPF-based keyword search for cloud storage," *IEEE Trans. Dependable Secure Comput.*, to be published, doi: 10.1109/TDSC.2023.3253786.
[27] 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 Trans. Inf. Forensics Secur.*, vol. 14, no. 4, pp. 870–885, Apr. 2019.
[28] G. Xu, H. Li, H. Ren, X. Lin, and X. Shen, "DNA similarity search with access control over encrypted cloud data," *IEEE Trans. Cloud Comput.*, vol. 10, no. 2, pp. 1233–1252, Second Quarter 2022.
[29] Y. Li et al., "DVREI: Dynamic verifiable retrieval over encrypted images," *IEEE Trans. Comput.*, vol. 71, no. 8, pp. 1755–1769, Aug. 2022.
[30] R. Li, A. X. Liu, Y. Liu, H. Xu, and H. Yuan, "Insecurity and hardness of nearest neighbor queries over encrypted data," in *Proc. IEEE 35th Int. Conf. Data Eng.*, 2019, pp. 1614–1617.
[31] B. Yao, F. Li, and X. Xiao, "Secure nearest neighbor revisited," in *Proc. IEEE 29th Int. Conf. Data Eng.*, 2013, pp. 733–744.
[32] X. Wang, J. Ma, X. Liu, and Y. Miao, "Search in my way: Practical outsourced image retrieval framework supporting unshared key," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 2485–2493.
[33] 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 Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5348–5359, Aug. 2021.
[34] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[35] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, 1999, pp. 223–238.
[36] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2015, pp. 1–34.
[37] X. Liu, R. Lu, J. Ma, L. Chen, and B. Qin, "Privacy-preserving patient-centric clinical decision support system on naive Bayesian classification," *IEEE J. Biomed. Health Informat.*, vol. 20, no. 2, pp. 655–668, Feb. 2015.
[38] S. Börzsönyi, D. Kossmann, and K. Stocker, "The skyline operator," in *Proc. IEEE 17th Int. Conf. Data Eng.*, 2001, pp. 421–430.
[39] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, "Toward efficient and privacy-preserving computing in Big Data era," *IEEE Netw.*, vol. 28, no. 4, pp. 46–50, 2014.
[40] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm," *Mach. Learn.*, vol. 37, no. 3, pp. 277–296, 1999.
[41] C. Zhang, L. Zhu, C. Xu, and R. Lu, "PPDP: An efficient and privacy-preserving disease prediction scheme in cloud-based E-healthcare system," *Future Gener. Comput. Syst.*, vol. 79, pp. 16–25, 2018.
[42] Z. Ma et al., "Lightweight privacy-preserving medical diagnosis in edge computing," *IEEE Trans. Serv. Comput.*, vol. 15, no. 3, pp. 1606–1618, May/Jun. 2022.
[43] Y. Sun, J. Liu, K. Yu, M. Alazab, and K. Lin, "PMRSS: Privacy-preserving medical record searching scheme for intelligent diagnosis in IoT healthcare," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1981–1990, Mar. 2022.
[44] M. Zhang, Y. Chen, and W. Susilo, "Decision tree evaluation on sensitive datasets for secure E-healthcare systems," *IEEE Trans. Dependable Secure Comput.*, to be published, doi: 10.1109/TDSC.2022.3219849.
[45] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, 1998, pp. 308–318.
[46] W. Ding, Z. Yan, and R. H. Deng, "Encrypted data processing with homomorphic re-encryption," *Inf. Sci.*, vol. 409, pp. 35–55, 2017.
[47] S. Xiang, F. Nie, and C. Zhang, "Learning a mahalanobis distance metric for data clustering and classification," *Pattern Recognit.*, vol. 41, no. 12, pp. 3600–3612, 2008.
[48] P. M. Roth, M. Hirzer, M. Köstinger, C. Beleznai, and H. Bischof, "Mahalanobis distance learning for person re-identification," in *Person Re-Identification*. Berlin, Germany: Springer, 2014, pp. 247–267.
[49] G. Wei, H. Cao, H. Ma, S. Qi, W. Qian, and Z. Ma, "Content-based image retrieval for lung nodule classification using texture features and learned distance metric," *J. Med. Syst.*, vol. 42, no. 1, pp. 1–7, 2018.
[50] H. Sarmadi, A. Entezami, B. Saeedi Razavi, and K.-V. Yuen, "Ensemble learning-based structural health monitoring by mahalanobis distance metrics," *Struct. Control Health Monit.*, vol. 28, no. 2, 2021, Art. no. e2663.
[51] S. Sun, "Segmentation-based adaptive feature extraction combined with mahalanobis distance classification criterion for heart sound diagnostic system," *IEEE Sensors J.*, vol. 21, no. 9, pp. 11009–11022, May 2021.

[52] O. Stan, M. Zayani, R. Sirdey, A. B. Hamida, A. F. Leite, and M. Mziou-Sallami, "A new crypto-classifier service for energy efficiency in smart cities," in *Proc. 7th Int. Conf. Smart Cities Green ICT Syst.*, 2018, pp. 78–88.
[53] Q. Wang, D. Zhou, Q. Guan, Y. Li, and J. Yang, "A privacy-preserving classifier in statistic pattern recognition," in *Proc. Cloud Comput. Secur.: 4th Int. Conf.*, 2018, pp. 496–507.
[54] M. Barbosa, D. Catalano, and D. Fiore, "Labeled homomorphic encryption - scalable and privacy-preserving processing of outsourced data," in *Proc. 22nd Eur. Symp. Res. Comput. Secur.*, 2017, pp. 146–166.

Dan Zhu received the BS degree from the School of Telecommunications Engineering, Xidian University, Xi'an, China, in 2017. She is currently working toward the PhD degree in Xidian University. Her research interests include applied cryptography, data security and privacy.

Hui Zhu (Senior Member, IEEE) received the BS and PhD degrees from Xidian University, Xi'an, China, in 2003 and 2009, respectively, and the MS 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.



Cheng Huang (Member, IEEE) received the BEng and MEng degrees in information security from Xidian University, China, in 2013 and 2016 respectively, and the PhD degree in electrical and computer engineering, University of Waterloo, ON, Canada, in 2020. He is currently a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, University of Waterloo. His research interests are in the areas of applied cryptography, cyber security and privacy in the mobile network.

Rongxing Lu (Fellow, IEEE) received the PhD degree from the Department of Electrical & Computer Engineering, University of Waterloo, Canada, in 2012. He is an associate professor with the Faculty of Computer Science, University of New Brunswick, Canada. Before that, he worked as an assistant professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore from April 2013 to August 2016. Rongxing Lu worked as a Postdoctoral Fellow with the University of Waterloo from May 2012 to April 2013. Currently, He 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 BS degree from Shaanxi Normal University, Xi'an, China, in 1988, the MS and PhD 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.

Xuemin (Sherman) Shen (Fellow, IEEE) received the PhD degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is currently a University professor with the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada. His research interests include network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular ad hoc and sensor networks. He is also a registered professional engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, Canadian Academy of Engineering Fellow, Royal Society of Canada Fellow, Chinese Academy of Engineering Foreign Member, and distinguished lecturer of the IEEE Vehicular Technology Society and Communications Society. He was the recipient of the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT) in 2021, R.A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society, Technical Recognition Award from Wireless Communications Technical Committee (2019) and AHSN Technical Committee (2013), Excellent Graduate Supervision Award in 2006 from the University of Waterloo, Premier's Research Excellence Award (PREA), and in 2003 from the Province of Ontario, Canada. He was the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, IEEE Globecom'07, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He is the president of the IEEE Communications Society. He was the vice president for Technical & Educational Activities, vice president for Publications, Member-at-Large on the Board of Governors, Chair of the Distinguished Lecturer Selection Committee, Member of IEEE Fellow Selection Committee of the ComSoc. He was also the editor-in-chief of the *IEEE IoT Journal*, *IEEE Network*, and *IET Communications*.