An Efficient and Privacy-Preserving Disease Risk Prediction Scheme for E-Healthcare

Xue Yang, Rongxing Lu, *Senior Member, IEEE*, Jun Shao, Xiaohu Tang, *Member, IEEE*, and Haomiao Yang, *Member, IEEE*

***Abstract*—Big data mining-driven disease risk prediction has become one of the important topics in the field of e-healthcare. However, without the security and privacy assurances, disease risk prediction cannot continue to flourish. To address this challenge, in this paper, an efficient and privacy-preserving disease risk prediction scheme for e-healthcare is proposed, hereafter referred to as EPDP. Compared with the up-to-date works, the proposed EPDP comprehensively achieves two phases of disease risk prediction, i.e., disease model training and disease prediction, while ensuring the privacy preservation. Specifically, a super-increasing sequence is combined with a homomorphic cryptographic algorithm to efficiently extract the symptom set of each disease in the phase of disease model training. Bloom filter technique is introduced to compute the prediction result in the phase of disease risk prediction. Besides, extensive performance evaluations demonstrate that our proposed EPDP attains outstanding efficiency advantage over the state-of-the-art in terms of both computational and communication overheads, and hence our EPDP is more suitable for real-time e-healthcare, especially medical emergency.**

***Index Terms*—Disease risk prediction, e-healthcare, homomorphic cryptographic algorithm, privacy-preserving.**

I. Introduction

With the rapid development of wireless sensors, smart devices and network technologies, the Internet of Things (IoT), as it can greatly improve the quality of life, has played an important role in the modern society [1]. As one of the major applications in IoT, e-healthcare has been widely researched since it has advantages in prevention and easy monitoring of diseases, ad hoc diagnosis and providing prompt medical attention in cases of accidents [2], [3]. E-healthcare includes many research fields, among which the extensive one is disease risk prediction as it can help to predict the disease risk and improve the diagnosis efficiency. Thus, in this paper, we focus on this popular research field.

In general, the disease risk prediction mainly consists of two phases: 1) disease model training and 2) remote disease prediction [4], [5]. In the phase of disease model training, a huge number of historical medical data containing patients' symptoms and confirmed diseases are collected by the resource-abundant third party, e.g., the cloud platform (CP), and then the training result is extracted from the collected data by means of big data mining technologies [6], [7]. After that healthcare providers (HPs), e.g., hospital or medical company, utilize the training result to predict the disease risk for undiagnosed patients based on the personal symptoms collected by medical monitoring devices or doctor visits. That is, in the whole process of disease risk prediction, confirmed patients provide their historical medical data for disease model training, while undiagnosed patients can use the disease prediction service to obtain the possible diseases by providing the collected symptoms. Unfortunately, as shown in most e-healthcare researches [8]–[11], security and privacy issues have significantly impeded the wide adoption of e-healthcare systems, since the exposure and abuse of personal health information (PHI) would bring about serious privacy leakage, let alone the involvement of not fully trusted third-party CPs [12], [13].

In principle, a promising disease risk prediction system should provide the following desirable properties.

1) *Comprehensiveness:* The disease risk prediction needs to provide the disease model training and remote disease prediction, simultaneously [14], [15].

Manuscript received August 5, 2018; revised September 30, 2018 and October 29, 2018; accepted November 11, 2018. Date of publication November 20, 2018; date of current version May 8, 2019. The work of X. Yang was supported in part by the 2017 Doctoral Innovation Fund Program of Southwest Jiaotong University under Grant D-CX201723, in part by the China Scholarship Council, and in part by the Major Frontier Project of Sichuan Province under Grant 2015JY0282. The work of R. Lu was supported in part by the NSERC Discovery Grant 04009 and in part by the NBIF Start-Up Grant Rif 2017-012, Grant HMF2017 YS-04, and Grant LMCRF-S-2018-03. The work of J. Shao was supported in part by ZJNSF Grant LZ18F020003 and in part by NSFC Grant 61472364. The work of X. Tang was supported by the Major Frontier Project of Sichuan Province under Grant 2015JY0282. The work of H. Yang was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0802003, in part by the National Natural Science Foundation of China under Grant U1633114, and in part by the Sichuan Science and Technology Program under Grant 2018GZ0202. *(Corresponding author: Rongxing Lu.)*

X. Yang is with the Information Security and National Computing Grid Laboratory, Southwest Jiaotong University, Chengdu 610031, China, and also with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada (e-mail: xueyang.swjtu@gmail.com).

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

J. Shao is with the School of Computer and Information Engineering, Zhejiang Gongshang University, Zhejiang 310018, China, and also with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada (e-mail: chn.junshao@gmail.com).

X. Tang is with the Information Security and National Computing Grid Laboratory, Southwest Jiaotong University, Chengdu 610031, China (e-mail: xhutang@swjtu.edu.cn).

H. Yang is with the School of Computer Science and Engineering, Center for Cyber Security, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: haomyang@uestc.edu.cn).

Digital Object Identifier 10.1109/JIOT.2018.2882224

2) *Privacy-Preservation:* The privacy-preservation is a decisive factor influencing the flourish of a disease risk prediction system [16], [17]. Naturally, if the privacy issue is not well addressed, confirmed patients would not like to provide their PHI for training. Meanwhile, undiagnosed patients would not use the prediction service.
3) *Efficiency:* Whether a disease risk prediction system can be applied to the practice greatly depends on the efficiency [18], [19]. For example, we consider an emergency scenario that a user suddenly fainted at home. In order to take proper first-aid measures, the personal symptoms collected by medical monitoring devices can be sent to the HP to obtain the disease prediction in time. Therefore, the response should be as fast as possible, i.e., the computational cost and communication overhead should be efficient.

Although many up-to-date disease risk prediction schemes have been proposed [20]–[22], they cannot meet all of the above requirements. Up to now, designing an efficient, comprehensive and privacy-preserving scheme for disease risk prediction remains a research challenge. Therefore, in this paper, we propose an efficient and privacy-preserving disease risk prediction scheme for e-healthcare, hereafter referred to as EPDP. The main contributions of this paper are three aspects.

1) We consider the comprehensive disease risk prediction system that includes the disease model training and remote disease prediction. Specifically, in the disease model training, we use the historical medical data collected from confirmed patients to train the naïve Bayesian classifier [23]. Then, the trained classifier can be used to extract the symptom vector set of each disease. Based on the extracted training results, our EPDP can help undiagnosed patients to predict the disease by using the efficient Bloom filter (BF) technique [24] in the phase of disease prediction.
2) Our EPDP achieves the privacy requirements of medical users (MUs) and the HP. In more details, since historical medical data are sensitive information for confirmed patients, each confirmed patient encrypts the historical medical data by the Okamoto–Uchiyama (OU) cryptosystem [25] before outsourcing. In the phase of disease prediction, in order to protect the privacy from disclosure, both the HP and undiagnosed patient use the keyed-cryptographic hash function to generate the BF and query element, respectively.
3) To ensure the efficiency of disease risk prediction system, our EPDP introduces a super-increasing sequence [26] to greatly reduce the encryption times and communication overhead. Specifically, instead of encrypting each dimension of multidimensional historical data one by one [27], in our EPDP, each confirmed patient can use this sequence to compress multidimensional historical data into 1-D, and then encrypts the compressed data by the OU cryptosystem. Besides, using the BF to complete the disease diagnosis can also greatly improve the efficiency. Further, extensive performance evaluations demonstrate that our EPDP attains outstanding efficiency advantage over the state-of-the-art in terms of both computational and communication overhead.

The remainder of this paper is organized as follows. We formalize the models and design goals in Section II. Then, we outline the definitions of OU cryptosystem and the BF technology in Section III. After that, we describe the proposed scheme in Section IV, followed by its privacy analysis and performance evaluation in Sections V and VI, respectively. We review some related works in Section VII. Finally, we draw our conclusions in Section VIII.

II. Models and Design Goals

In this section, we formalize the system model and threat model used in this paper and identify our design goals.

A. System Model

The system model (see Fig. 1) is compromised of a CP, a number of MUs and an HP. Similar to [27], we assume that there are n_s symptom attributes $(X_1, X_2, \ldots, X_{n_s})$ and n_d disease classes $(Y_1, Y_2, \ldots, Y_{n_d})$ in the system. The role of each entity is described as follows.

1) *MUs:* MUs act as either confirmed patients or undiagnosed patients. For each confirmed patient u_i, he or she has an n_s-dimensional symptom vector $\mathbf{x}^i = (x_1^i, \ldots, x_{n_s}^i)$ and the corresponding n_d-dimensional confirmed disease vector $\mathbf{y}^i = (y_1^i, \ldots, y_{n_d}^i)$, where $x_j^i, y_k^i \in \{0, 1\}$ for $j = 1, \ldots, n_s$ and $k = 1, \ldots, n_d$. In particular, $x_j^i = 1$ means that u_i has the symptom attribute X_j, and $x_j^i = 0$ otherwise. $y_k^i = 1$ means that u_i suffers from the disease Y_k, and $y_k^i = 0$ otherwise. Besides, in order to help the HP train the naïve Bayesian classifier, u_i needs to generate an $n_s n_d$-dimensional vector $\mathbf{z}^i = (z_{11}^i, z_{21}^i, \ldots, z_{n_s 1}^i, \ldots, z_{n_s n_d}^i)$, where $z_{jk}^i = x_j^i \cdot y_k^i$. Thus, each confirmed patient u_i needs to provide the historical medical data $\mathbf{x}^i$, $\mathbf{y}^i$, and $\mathbf{z}^i$ to the HP for disease model training. As an undiagnosed patient, he or she has an undiagnosed symptom vector $\mathbf{b} = (b_1, \ldots, b_{n_s})$ collected by body medical sensors and wants to obtain the prediction result regarding $\mathbf{b}$ from the CP.
2) *CP:* CP is responsible for helping the HP collect the historical medical data from confirmed patients while predicting the disease for undiagnosed patients on the behalf of the HP.
3) *HP:* HP, e.g., a hospital or a medical company, provides the service of disease risk prediction. Specifically, with the help of the CP, the HP can obtain the training data of the naïve Bayesian classifier, which can be used to extract the symptom set of each disease including all symptom vectors that may cause patients to suffer from the corresponding disease. Considering the benefit of the CP, the HP also would like to delegate the CP to efficiently predict diseases and return the result to undiagnosed patients.

As shown in Fig. 1, the overall workflow of our scheme mainly contains two phases. The first phase is the disease model training, where the HP delegates the CP to collect the

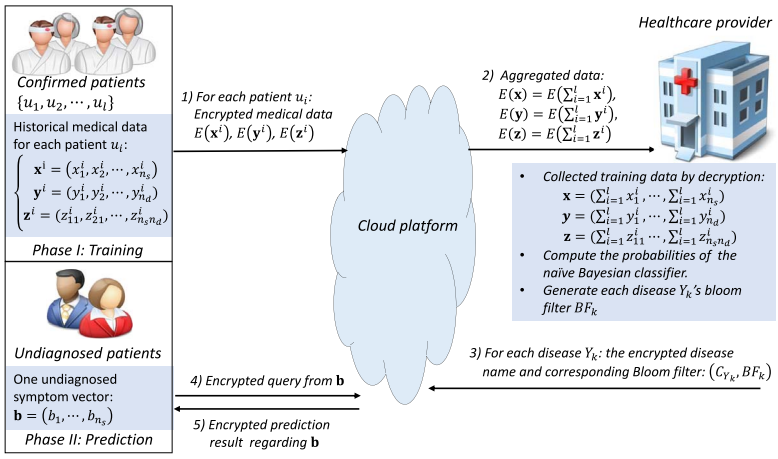


Fig. 1. System model under consideration.

historical medical data from confirmed patients, and then trains the naïve Bayesian classifier. The second phase is the disease risk prediction, where the HP outsources the prediction service to the CP to help undiagnosed patients predict the possible diseases. Specifically, in the first phase, each confirmed patient u_i encrypts the historical medical data $\mathbf{x}^i$, $\mathbf{y}^i$, and $\mathbf{z}^i$, and then sends the encrypted data to the CP. After receiving l historical medical data, the CP performs the aggregation operation and forwards the aggregated data to the HP. Finally, the HP decrypts the received data to compute the probabilities of the naïve Bayesian classifier. After obtaining the probabilities, the HP generates the BF_k for each disease Y_k, and then outsources BF_k together with the encrypted name C_{Y_k} to the CP for risk prediction. In the second phase, an undiagnosed patient generates the encrypted query based on the undiagnosed symptom vector $\mathbf{b}$ and sends it to the CP for predicting the possible diseases. Based on the received query, the CP can judge whether this patient suffers from the disease C_{Y_k} by performing the membership query of BF_k. Finally, the CP returns the encrypted prediction result including the possible disease names to this patient.

B. Threat Model

In our threat model, the CP is considered as honest-but-curious, which strictly follows the underlying scheme, but is interested in the privacy of MUs and the HP. Similar to [27] and [28], we consider both HP and MUs as honest-but-curious. Specifically, the HP provides the correct information for disease risk prediction, but is curious about the privacy of MUs, i.e., historical medical data or predicted disease. If the HP obtains a patient's disease history, in addition to using them for disease model training, it is likely to obtain monetary benefits by selling that information to some related companies, e.g., the insurance company and employers. MUs provide correct medical data for disease model training or disease risk prediction, but also attempt to know the training results, which is regarded as intellectual properties of the HP. Besides, MUs also attempt to know other patients' medical data with the same reason as the HP and CP.

Note that there may be other attacks and security requirements, e.g., collusion attack and access control, in e-healthcare. Since our objective is on the privacy-preserving disease risk prediction, those attacks and security requirements are currently out of scope for this article and will be considered in future work.

C. Design Goals

Under the aforementioned models, the design goals of our EPDP are described as follows.

1) *Privacy-Preservation:* Any adversary including the CP, the HP or MUs cannot feasibly obtain the sensitive data of other entities based on the obtained data. Specifically, as the profit company, the training results are considered as the HP's own asset, which should be protected from disclosing. In other words, although the CP can perform the disease risk prediction on the behalf of the HP, it cannot obtain the training results. Besides, even though MUs provide medical data, they cannot obtain the training results from disease model training or disease risk prediction. For MUs, the medical data (especially the suffered disease) are extremely sensitive information, thus they may refuse to provide the medical data or use the service of disease prediction without the good protection of privacy. That is, the proposed system should also achieve the privacy preservation for MUs.
2) *Efficiency:* Although the CP has the powerful computational capacity to deal with time-consuming calculations, the computational efficiency is still expected to be improved for time-sensitive e-healthcare, especially in the case of the medical emergency that needs to retrieve diagnosis results in time. Meanwhile, our EPDP should ensure that the CP can independently make the judgment without interaction. Moreover, the HP and MUs outsource time-consuming calculations to the CP, but before outsourcing the data, they have to perform some calculations to protect the privacy. Hence, the corresponding computing should be efficient, especially for the capacity-limited mobile MUs. Besides, the communication overhead is an important factor influencing the delay, thus we need to reduce the communication overheads as much as possible.

III. Preliminary

In this section, we outline the OU cryptosystem [25] and BF [24], which serve as the building blocks of our EPDP. The details are shown as follows.

A. Okamoto–Uchiyama Cryptosystem

The OU cryptosystem mainly includes three algorithms: 1) key generation; 2) encryption; and 3) decryption.

1) *Key Generation:* Given the security parameter κ, choose two large primes p and q with the same bit-length $|p| = |q| = \kappa$, and compute $N = p^2q$. Then, choose $g \in \mathbb{Z}_N^*$ such that the order of $g^{p-1} \bmod p^2$ is p, and set $g_1 = g^N \bmod N$. The public key is $pk = (N, g, g_1, \kappa)$ and the corresponding private key is $sk = (p, q)$.
2) *Encryption:* Given the message $0 \le m < 2^{\kappa-1}$, choose a random number $r \in \mathbb{Z}_N$, then the ciphertext can be computed as

$$C = E(m) = g^m \cdot g_1^r \bmod N. \tag{1}$$

3) *Decryption:* For a ciphertext $C \in \mathbb{Z}_N$, the message m can be recovered with the private key as

$$D(C) = \left(\frac{\left(C^{p-1} \bmod p^2\right) - 1}{p} \right) \cdot \alpha^{-1} \bmod p \tag{2}$$

where $\alpha = [((g^{p-1} \bmod p^2) - 1)/p] \bmod p$. The correctness of the OU cryptosystem can be referred to [25].

Besides, the OU cryptosystem supports the additive homomorphism

$$\begin{aligned} D(E(m_1) \cdot E(m_2) \bmod N) &= D\left(g^{m_1+m_2} g_1^{r_1+r_2} \bmod N\right) \\ &= D(E(m_1 + m_2)) \end{aligned}$$

where $m_1 + m_2 < 2^{\kappa-1}$.

Note that, in addition to the OU cryptosystem, the Paillier cryptosystem [29] has also been widely applied in ciphertext-based operation applications [30]–[32]. However, with the same security parameter, e.g., $|p| = |q| = 512$ bits, the message space and ciphertext space of Paillier cryptosystem are, respectively, 1024 bits and 2048 bits, while the message space and ciphertext space of the OU cryptosystem are, respectively, around 512 bits and 1536 bits. As a result, for some applications with small space, it is better to choose the OU cryptosystem. In our scheme, since the plaintext is relatively small, we adopt the OU cryptosystem, which can reduce the encryption and decryption costs as well as the communication overhead.

B. Bloom Filter

The BF is a space-efficient data structure for representing a set and testing whether an element is definitely not or possibly in this set. Specifically, a BF is initialized by InitBF(L) to generate an array of L bits, where all bits are set to 0 (see Fig. 2).

The BF mainly contains two operations: 1) element addition and 2) membership query. Specifically, to add an element or query whether an element is in the set, the BF chooses f independent hash functions $\{h_1, h_2, \ldots, h_f\}$, each of which uniformly maps the element to one of L array positions, i.e., $h_i : \{0, 1\}^* \to \{1, 2, \ldots, L\}$ for $i = 1, 2, \ldots, f$.

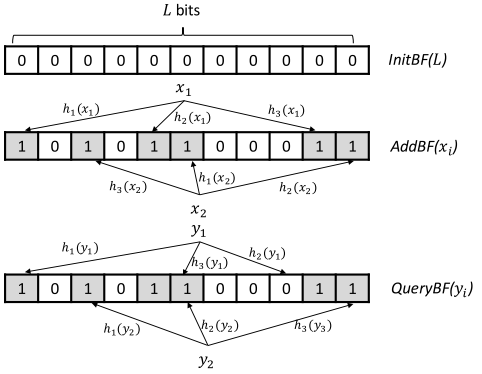


Fig. 2. Example of a BF initialized by InitBF(11) and using three hash functions $\{h_1, h_2, h_3\}$. Each element x_i can be added in the set by executing AddBF(x_i). We can check whether an element y_i is in the set by calling QueryBF(y_i). Specifically, since a 0 is found at the ninth bit, y_1 cannot be in the set. The element y_2 is either in the set or the filter has yielded a false positive.

1) *Element Addition AddBF(x):* In order to add an element x in a set, f array positions in bit array are computed as $\{h_1(x), h_2(x), \ldots, h_f(x)\}$. Then, set the $h_i(x)$-th bit in the array to 1 for $i = 1, 2, \ldots, f$. It is worth noting that a bit location can be set to 1 multiple times, but only the first change has an effect.
2) *Membership Query QueryBF(y):* To query whether an element y is included in the set, check the value of the $h_i(y)$-th bit in the array for $i = 1, 2, \ldots, f$. The result of QueryBF(y) is either 1 or 0, i.e., QueryBF(y) $\to \{0, 1\}$. Specifically, if any of the bits at f positions is 0, then return 0, which means that the element y is definitely not in the set. If all are 1, then return 1, which means that either the element y is in the set, or the bits have by chance been set to 1 during the addition of other elements, resulting in a false positive [33].

Particularly, if n elements have been added into the BF, and each element is mapped to the f positions with equal probability, the false positive probability $\mathcal{P}$ is calculated as

$$\mathcal{P} = \left(1 - (1 - 1/L)^{fn}\right)^f \approx \left(1 - e^{-fn/L}\right)^f \tag{3}$$

which can be minimized when $f = (L/n) \ln 2$. Fig. 2 provides an example of BF.

IV. Proposed Scheme

In this section, we present our EPDP, which mainly consists of the following three phases: 1) system initialization; 2) disease model training; and 3) disease risk prediction. Before that, we would like to describe the notations in the proposed scheme in Table I.

A. System Initialization

The HP initializes the system as follows.

1) Take the security parameter κ_0 as input, and output the OU parameters $(N, g, g_1, \kappa_0, p, q)$ by running the corresponding key generation algorithm, where the public key is $pk = (N, g, g_1, \kappa_0)$ and the corresponding private key is $sk = (p, q)$.

TABLE I
NOTATIONS USED IN THE PROPOSED SCHEME

Notation	Description
$pk = (N, g, g_1, \kappa_0)$	the HP's public keys of the OU cryptosystem
$sk = (p, q)$	the HP's private keys of the OU cryptosystem
$SE(\lambda, \cdot)$	symmetric encryption algorithm with the key λ, i.e., AES used in this paper
$H(\lambda, \cdot)$	cryptographic hash function with key λ
L	the size of the Bloom filter
$\{h_1, \ldots, h_f\}$	f hash functions in the Bloom filter
$E(\cdot)$	encryption algorithm of the OU cryptosystem
$D(\cdot)$	decryption algorithm of the OU cryptosystem
$\{u_1, \ldots, u_l\}$	l confirmed patients in the system
$Y_1, Y_2, \ldots, Y_{n_d}$	n_d disease classes in the system
$X_1, X_2, \ldots, X_{n_s}$	n_s symptom attributes in the system
$\mathbf{x}^i = (x_1^i, \ldots, x_{n_s}^i)$	symptom vector of the confirmed patient u_i
$\mathbf{y}^i = (y_1^i, \ldots, y_{n_d}^i)$	disease vector of the confirmed patient u_i
BF_k, $k = 1, \ldots, n_d$	Bloom filter corresponding to the disease Y_k
$InitBF_k(L)$	the initialization operation for BF_k
$AddBF_k(\cdot)$	the element addition operation in BF_k
$QueryBF_k(\cdot)$	the element query operation in BF_k

Note that in this paper, the lowercase letters in bold represent vectors.

2) Choose a symmetric encryption algorithm, e.g., AES-256 in this paper, where the symmetric key λ is randomly chosen from the key space $\lambda \in \{0, 1\}^{\kappa_1}$.
3) Choose a keyed-cryptographic hash function $H(\lambda, \cdot) : \{0, 1\}^* \rightarrow \{0, 1\}^{\kappa_2}$, where κ_2 is the bit length of the hash value. To this end, we can use keyed-hashing for message authentication code (HMAC) technique [34].
4) Choose a super-increasing sequence vector $\mathbf{a} = (a_1 = 1, a_2, \ldots, a_{n_s n_d})$, where $a_2, \ldots, a_{n_s n_d}$ are integers such that $\sum_{j=1}^{i-1} a_j \cdot l < a_i$ for $i = 2, \ldots, n_s n_d$, and $\sum_{i=1}^{n_s n_d} a_i \cdot l < 2^{\kappa - 1}$. Note that n_s, n_d, and l represent the number of symptom attributes, disease classes, and confirmed patients, respectively. With this sequence, we can compress multidimensional data into the 1-D, and then reconstruct each dimension of the multidimensional data from this compressed data. Specifically, in the disease model training, each confirmed patient uses this sequence to compress the multidimensional historical data into 1-D and encrypts the compressed data by the OU cryptosystem, which can greatly reduce the encryption times and the corresponding communication overhead.
5) Set an appropriate value for the BF's length L and choose f independent hash functions $\{h_1, h_2, \ldots, h_f\}$ such that $h_i : \{0, 1\}^* \rightarrow \{1, 2, \ldots, L\}$ for $i = 1, 2, \ldots, f$. For each disease Y_k, initialize the corresponding filter BF_k by executing $\mathrm{InitBF}_k(L)$.

In the end, HP publishes the system parameters as $(N, g, g_1, \kappa_0, \mathbf{a}, H, h_1, h_2, \ldots, h_f)$ and keeps (p, q, λ) secret.

B. Privacy-Preserving Disease Model Training

In this section, we introduce the details about the privacy-preserving disease model training, which mainly contains four parts: 1) historical data encryption; 2) ciphertext aggregation; 3) decryption; and 4) extraction of the symptom set. Specifically, l confirmed patients $\{u_1, u_2, \ldots, u_l\}$ encrypt the historical medical data by the OU encryption algorithm $E(\cdot)$, and send these encrypted data to the CP. After receiving these encrypted data, the CP aggregates them and forwards the aggregated data to the HP for computing the probabilities of naïve Bayesian classifier. Based on the trained results, i.e., the probabilities of naïve Bayesian classifier, the HP can obtain the BF of each disease, which represents the set containing all symptom vectors that may cause the corresponding disease. The details are shown below.

1) Historical Data Encryption: As described in Section II-A, each confirmed patient u_i, $i = 1, 2, \ldots, l$, needs to provide three vectors $\mathbf{x}^i = (x_1^i, \ldots, x_{n_s}^i)$, $\mathbf{y}^i = (y_1^i, \ldots, y_{n_d}^i)$ and $\mathbf{z}^i = (z_{11}^i, z_{21}^i, \ldots, z_{n_s 1}^i, \ldots, z_{n_s n_d}^i)$. In order to protect the privacy, u_i generates the ciphertexts with the HP's public parameters $(N, g, g_1, \kappa_0, \mathbf{a})$ as follows.

1) u_i uses the super-increasing sequence vector $\mathbf{a}$ to compress $\mathbf{x}^i$, $\mathbf{y}^i$, and $\mathbf{z}^i$ into three plaintexts as follows:

$$M_1^i = a_1 x_1^i + a_2 x_2^i + \cdots + a_{n_s} x_{n_s}^i \tag{4}$$

$$M_2^i = a_1 y_1^i + a_2 y_2^i + \cdots + a_{n_d} y_{n_d}^i \tag{5}$$

$$M_3^i = a_1 z_{11}^i + a_2 z_{21}^i + \cdots + a_{n_s n_d} z_{n_s n_d}^i \tag{6}$$

where $M_1^i, M_2^i, M_3^i < 2^{\kappa_0 - 1}$, and the correctness of message space will be described in Section IV-B3.

2) u_i encrypts these three plaintexts by calling the encryption algorithm $E(\cdot)$

$$C_1^i = g^{M_1^i} \cdot g_1^{r_1^i} \bmod N \tag{7}$$

$$C_2^i = g^{M_2^i} \cdot g_1^{r_2^i} \bmod N \tag{8}$$

$$C_3^i = g^{M_3^i} \cdot g_1^{r_3^i} \bmod N \tag{9}$$

where $r_1^i, r_2^i, r_3^i \in \mathbb{Z}_N$ are random numbers.

3) u_i sends (C_1^i, C_2^i, C_3^i) to the CP.

2) Ciphertext Aggregation: After receiving (C_1^i, C_2^i, C_3^i) from l confirmed patients, the CP aggregates them as follows:

$$\begin{aligned} C_1 &= \prod_{i=1}^{l} C_1^i \\ &= g^{\sum_{i=1}^{l} M_1^i} \cdot g_1^{\sum_{i=1}^{l} r_1^i} \bmod N \\ &= g^{a_1 \sum_{i=1}^{l} x_1^i + \cdots + a_{n_s} \sum_{i=1}^{l} x_{n_s}^i} \cdot g_1^{\sum_{i=1}^{l} r_1^i} \bmod N \\ C_2 &= \prod_{i=1}^{l} C_2^i \\ &= g^{\sum_{i=1}^{l} M_2^i} \cdot g_1^{\sum_{i=1}^{l} r_2^i} \bmod N \\ &= g^{a_1 \sum_{i=1}^{l} y_1^i + \cdots + a_{n_d} \sum_{i=1}^{l} y_{n_d}^i} \cdot g_1^{\sum_{i=1}^{l} r_2^i} \bmod N \\ C_3 &= \prod_{i=1}^{l} C_3^i \\ &= g^{\sum_{i=1}^{l} M_3^i} \cdot g_1^{\sum_{i=1}^{l} r_3^i} \bmod N \\ &= g^{a_1 \sum_{i=1}^{l} z_{11}^i + \cdots + a_{n_s n_d} \sum_{i=1}^{l} z_{n_s n_d}^i} \cdot g_1^{\sum_{i=1}^{l} r_3^i} \bmod N. \end{aligned}$$

Then, the CP sends (C_1, C_2, C_3) to the HP.

3) Decryption: After receiving (C_1, C_2, C_3), the HP performs the following steps to read the aggregated and encrypted

Algorithm 1: Recover the Aggregated Report

Input: $M = a_1m_1 + a_2m_2 + \cdots + a_nm_n$ and a super-increasing sequence $\mathbf{a} = (a_1 = 1, \cdots, a_n)$ with $\sum_{j=1}^{i-1} a_jm_j < a_i$ for $i = 2, \ldots, n$.

Output: $(m_1, m_2, \ldots, m_n)$.

Set $t_n = M$;

for $i = n$ *to* 2 **do**

  $t_{i-1} = t_i \bmod a_i$;

  $m_i = \frac{t_i - t_{i-1}}{a_i}$;

$m_1 = t_1$;

return $(m_1, m_2, \ldots, m_n)$;

report (C_1, C_2, C_3). In order to facilitate the description, we use C_1 as an example. Specifically, C_1 is formed by

$$C_1 = g^{a_1 \sum_{i=1}^{l} x_1^i + \cdots + a_{n_s} \sum_{i=1}^{l} x_{n_s}^i} \cdot g_1^{\sum_{i=1}^{l} r_1^i} \bmod N.$$

Step 1: By taking $M_1 = a_1 \sum_{i=1}^{l} x_1^i + \cdots + a_{n_s} \sum_{i=1}^{l} x_{n_s}^i$ and $r_1 = \sum_{i=1}^{l} r_1^i$, the report $C_1 = g^{M_1} g_1^{r_1} \bmod N$ is still a ciphertext of OU cryptosystem. Therefore, the HP runs $D(\cdot)$ algorithm with the private key $sk = (p, q)$ to recover M_1 as

$$M_1 = D(C_1) = \left(\frac{\left(C_1{}^{p-1} \bmod p^2\right) - 1}{p} \right) \cdot \alpha^{-1} \bmod p$$

where $\alpha = [((g^{p-1} \bmod p^2) - 1)/p] \bmod p$.

Step 2: Given the inputs $(a_1, a_2, \ldots, a_{n_s})$ and M_1, by invoking Algorithm 1, the HP can recover and store the aggregated data $(x_1, x_2, \ldots, x_{n_s})$, where $x_j = \sum_{i=1}^{l} x_j^i$ for $j = 1, 2, \ldots, n_s$.

The Correctness of Data Recovery: We analyze the correctness of both steps as follows.

1) *The Correctness of Step 1:* Since $x_j^i \in \{0, 1\}$, $x_j = \sum_{i=1}^{l} x_j^i \le l$ for $j = 1, 2, \ldots, n_s$. We can obtain

$$\begin{aligned} M_1 &= a_1x_1 + a_2x_2 + \cdots + a_{n_s}x_{n_s} \\ &\le a_1l + a_2l + \cdots + a_{n_s}l \\ &= \sum_{i=1}^{n_s} a_il. \end{aligned}$$

As defined in Section IV-A, $\sum_{j=1}^{i-1} a_jl < a_i$ for $i = 2, \ldots, n_sn_d$, and $\sum_{i=1}^{n_sn_d} a_il < 2^{\kappa_0 - 1}$, so we have $\sum_{i=1}^{n_s} a_il < a_{n_s+1} < \sum_{i=1}^{n_sn_d} a_il < 2^{\kappa_0-1}$. That is, the data M_1 meets the message space of encryption algorithm $E(\cdot)$, and can be correctly decrypted by running $D(\cdot)$.

2) *The Correctness of Step 2:* Since $\sum_{j=1}^{i-1} a_jx_j < \sum_{j=1}^{i-1} a_j \cdot l < a_i$, based on the correctness analysis of Algorithm 1, the data $(x_1, x_2, \ldots, x_{n_s})$ can be correctly recovered.

Similarly, the HP can obtain the aggregated data $(y_1, y_2, \ldots, y_{n_d})$ and $(z_{11}, z_{12}, \ldots, z_{n_sn_d})$ from C_2 and C_3, respectively, where $y_k = \sum_{i=1}^{l} y_k^i$ and $z_{jk} = \sum_{i=1}^{l} x_j^i \cdot y_k^i$, for $j = 1, \ldots, n_s,\ k = 1, \ldots, n_d$.

The Correctness of Algorithm 1: In Algorithm 1, $t_n = M$, since $\sum_{j=1}^{i-1} a_jm_j < a_i$, we have

$$a_1m_1 + a_2m_2 + \cdots + a_{n-1}m_{n-1} < a_n.$$

Therefore, $t_{n-1} = t_n \bmod a_n = a_1m_1 + \cdots + a_{n-1}m_{n-1}$, and

$$\frac{t_n - t_{n-1}}{a_n} = \frac{a_nm_n}{a_n} = m_n.$$

With the similar procedure, we can also prove each m_i for $i = 1, 2, \ldots, n-1$. The details can be referred to [26].

4) Extraction of Symptom Vector Set: In this section, we present the details about how to extract the symptom vector set including all symptom vectors that may cause patients to suffer from the corresponding disease. It is worth noting that this operation is performed by the HP, so all data is processed in plaintext form.

With the data $(x_1, x_2, \ldots, x_{n_s})$, $(y_1, y_2, \ldots, y_{n_d})$, and $(z_{11}, z_{12}, \ldots, z_{n_sn_d})$, the HP computes probabilities of the naïve Bayesian classifier as follows:

$$\begin{aligned} \Pr(X_j = 1 | Y_k = 1) &= \frac{z_{jk}}{y_k} \\ \Pr(X_j = 1 | Y_k = 0) &= \frac{x_j - z_{jk}}{l - y_k} \\ \Pr(Y_k = 1) &= \frac{y_k}{l} \\ \Pr(Y_k = 0) &= 1 - \Pr(Y_k = 1) \\ \Pr(X_j = 0 | Y_k = 1) &= 1 - \Pr(X_j = 1 | Y_k = 1) \\ \Pr(X_j = 0 | Y_k = 0) &= 1 - \Pr(X_j = 1 | Y_k = 0) \end{aligned}$$

where $\{X_1, \ldots, X_{n_s}\}$ and $\{Y_1, \ldots, Y_{n_d}\}$ denote n_s symptom attributes and n_d disease types, respectively. Additionally, $X_j = 1$ denotes a patient satisfies the symptom X_j, and $X_j = 0$ otherwise. $Y_k = 1$ denotes a patient suffers from the disease Y_k, and $Y_k = 0$ otherwise. Note that $\Pr(X_j|Y_k)$ and $\Pr(Y_k)$, where $j = 1, 2, \ldots, n_s$ and $k = 1, 2, \ldots, n_d$, should be kept privately by HP as its own asset.

For each disease Y_k, $k = 1, 2, \ldots, n_d$, the HP can generate the corresponding BF_k by executing Algorithm 2. Specifically, when the dimension of the binary vector is n_s, there exist 2^{n_s} possible binary vectors in total, denoted as $\mathbf{w}_i = (w_{i1}, w_{i2}, \ldots, w_{in_s})$, $i = 1, 2, \ldots, 2^{n_s}$. For each binary vector $\mathbf{w}_i = (w_{i1}, \ldots, w_{in_s})$, based on the training results $\Pr(X_j|Y_k)$ and $\Pr(Y_k)$, $j = 1, 2, \ldots, n_s$, $k = 1, 2, \ldots, n_d$, and Bayes's theorem [35], the HP can use (10) and (11) to compute the probability of having the disease Y_k and the probability without suffering from the disease Y_k, respectively, i.e., β_1^k and β_0^k. If $\beta_1^k > \beta_0^k$ (i.e., the probability of having the disease is larger than the probability without suffering the disease), then it implies that a patient with $\mathbf{w}_i$ may suffers from the disease Y_k. Thus, $\mathbf{w}_i$ can be added to the BF_k by executing $\mathrm{AddBF}_k(H(\lambda, \mu_i))$, where λ is the HP's private key and μ_i is the decimal number of the binary vector $\mathbf{w}_i$. After the HP has tried 2^{n_s} binary vectors, it can obtain the final BF_k that includes all symptom vectors corresponding to the disease Y_k.

According to each BF_k for $k = 1, 2, \ldots, n_d$, we can see that determining whether an undiagnosed patient may suffer from the disease Y_k is equivalent to querying whether an element (i.e., undiagnosed vector in this paper) is in the set (i.e., symptom vector set BF_k). In other words, the disease diagnosis is equivalent to the membership query in the BF. Next, in order to save the overheads of storage and computation, the HP would like to delegate the CP to efficiently diagnose diseases and return the result to undiagnosed patients on his behalf. To this end, besides the BF_k, the HP also needs to outsource the corresponding disease name Y_k to the CP. In order to protect the privacy, the HP encrypts the disease name Y_k

Algorithm 2: Extraction of Symptom Vector Set

Input: All n_s-dimensional binary vectors $\mathbf{w}_i = (w_{i1}, \ldots, w_{in_s})$, $i = 1, 2, \ldots, 2^{n_s}$.

Output: Bloom filter BF_k, which contains all symptom vectors that may cause patients to suffer from the disease Y_k.

Initialize BF_k by executing $InitBF_k(L)$;

for $i = 1$ *to* 2^{n_s} **do**

compute the probability of having the disease Y_k:

$$\beta_1^k = \prod_{j=1}^{n_s} \Pr\big(w_{ij} = X_j | Y_k = 1\big) \cdot \Pr(Y_k = 1) \tag{10}$$

where $\Pr\big(w_{ij} = X_j | Y_k = 1\big) = w_{ij} \Pr(X_j = 1 | Y_k = 1) + (1 - w_{ij}) \Pr(X_j = 0 | Y_k = 1)$;

compute the probability without suffering from the disease Y_k:

$$\beta_0^k = \prod_{j=1}^{n_s} \Pr\big(w_{ij} = X_j | Y_k = 0\big) \cdot \Pr(Y_k = 0) \tag{11}$$

where $\Pr\big(w_{ij} = X_j | Y_k = 0\big) = w_{ij} \Pr(X_j = 1 | Y_k = 0) + (1 - w_{ij}) \Pr(X_j = 0 | Y_k = 0)$;

if $\beta_1^k > \beta_0^k$ **then**

compute $\mathbf{w}_i$'s decimal value μ_i ;

based on the private key λ, compute $H(\lambda, \mu_i)$;

execute $AddBF_k(H(\lambda, \mu_i))$;

return Bloom filter BF_k of the disease Y_k;

before outsourcing as

$$C_{Y_k} = \mathrm{SE}(\lambda, Y_k) \tag{12}$$

where $\mathrm{SE}(\lambda, \cdot)$ is the symmetric encryption algorithm with the key λ, e.g., AES algorithm used in this paper.

Finally, the HP outsources each tuple (BF_k, C_{Y_k}), $k = 1, 2, \ldots, n_d$, to the CP for storing and disease prediction.

5) Data Update Discussion: In the practical scenario, data update is frequent and can help improve the accuracy of the classifier. Our scheme can also deal with the data update. Specifically, the CP initially sets an appropriate value for l, and then performs the aggregation each time after receiving l historical data. Note that l is usually set to be relatively large, since it will hardly affect the update of the training model when l is small. In other words, if the CP only received one historical data, it will not forward it to the HP, which can also help to protects the privacy of the individual confirmed patient. After receiving the new aggregated data, the HP recalculates the probabilities of naïve Bayesian classifier, and then updates the BF_k for each disease Y_k based on Algorithm 2. Since the HP performs these update operations in the plaintext domain, the computational efficiency can be guaranteed.

C. Privacy-Preserving Disease Risk Prediction

In this section, we describe the privacy-preserving disease risk prediction, which includes four parts: 1) member registration; 2) encrypted query generation; 3) disease risk diagnosis; and 4) prediction result retrieving. Before that, in order to prevent the medical data of an undiagnosed patient from being guessed by other patients, the CP chooses a secure public-key cryptosystem (e.g., RSA-OAEP cryptosystem [36]), where $(\mathrm{PK_{CP}}, \mathrm{SK_{CP}})$ is the public and private key pair, $E_{\mathrm{PK_{CP}}}(\cdot)$ and $D_{\mathrm{SK_{CP}}}(\cdot)$ are the corresponding encryption and decryption algorithms, respectively. The details are shown as follows.

1) Member Registration: Suppose an undiagnosed user u_c has the vector $\mathbf{b} = (b_1, b_2, \ldots, b_{n_s})$, where $b_i \in \{0, 1\}$ for $i = 1, 2, \ldots, n_s$, and wants to know the possible diseases. In order to leverage the service of disease risk prediction provided by the HP, u_c first needs to register as a member of the HP. Then, the HP would send the secret key λ to the registered member u_c by the secure channel.

2) Encrypted Query Generation: With the obtained secret key λ, u_c can compute the query element as $H(\lambda, \varphi)$, where φ is the decimal value of the undiagnosed binary vector $\mathbf{b}$. Then, u_c randomly chooses a κ_1-bit string K_{u_c}, i.e., $K_{u_c} \in \{0, 1\}^{\kappa_1}$, and computes the encrypted query as

$$C_{u_c} = E_{\mathrm{PK_{CP}}}\big(H(\lambda, \varphi) \| K_{u_c}\big).$$

Note that K_{u_c} is a secret key selected by u_c to protect the prediction result from disclosure. Finally, u_c sends C_{u_c} to the CP for disease risk diagnosis.

3) Disease Risk Diagnosis: After receiving the C_{u_c}, the CP can obtain the query element $H(\lambda, \varphi)$ and the u_c's secret key K_{u_c} by performing the decryption algorithm $D_{\mathrm{SK_{CP}}}(C_{u_c})$. Then, with the query element $H(\lambda, \varphi)$, the CP obtains the prediction result by executing Algorithm 3. More precisely, for each tuple (BF_k, C_{Y_k}), $k = 1, 2, \ldots, n_d$, the CP checks whether u_c may suffer from the corresponding disease by performing the membership query operation $\mathrm{QueryBF}_k(H(\lambda, \varphi))$.

a) If $\mathrm{QueryBF}_k(H(\lambda, \varphi)) \to 0$, it means that φ is not in BF_k. Since each binary vector corresponds to the unique decimal value, $\mathbf{b}$ is not included in the set represented by BF_k. That is, u_c may not suffer from this disease.

b) If $\mathrm{QueryBF}_k(H(\lambda, \varphi)) \to 1$, it implies that u_c may suffer from this disease. Thus, the CP adds the encrypted disease name C_{Y_k} to the diagnosis result set S. Note that in this case, the BF may yield a false positive result, but we can minimize it by setting the appropriate parameters L and f [see (3)].

After checking for n_d diseases, the CP obtains the final prediction result set S including all encrypted disease names that u_c may suffer. In order to prevent the content of the encrypted disease name in S from being guessed by other patients, the CP encrypts the set S with u_c's secret key K_{u_c} as follows.

a) For each $C_{Y_k} \in S$, compute

$$C_{Y_k}^* = \mathrm{SE}\big(K_{u_c}, C_{Y_k}\big)$$

where $\mathrm{SE}(K_{u_c}, \cdot)$ is the symmetric encryption algorithm with the key K_{u_c}, e.g., AES algorithm. We use S^* to represent the encrypted set S. Then, the CP returns the set S^* to u_c.

In addition, the CP can help the HP to collect the statistics based on the prediction results. Specifically, the CP can know how many undiagnosed patients are likely to suffer from the disease labeled k. For example, if the encrypted name C_{Y_k} is

Algorithm 3: Disease Prediction Algorithm

Input: The query element $H(\lambda, \varphi)$ and $\left(BF_k, C_{Y_k}\right)$ for $k = 1, 2, \ldots, n_d$.
Output: Diagnosis result set S that includes all possible diseases.
Initialize S to an empty set, i.e., $S = \emptyset$;
for $k = 1$ *to* n_d **do**
  execute $QueryBF_k(H(\lambda, \varphi))$;
  if $QueryBF_k(H(\lambda, \varphi)) \rightarrow 1$ **then**
    add C_{Y_k} to the set S, i.e., $S = S \cup \{C_{Y_k}\}$;
return the set S;

included in a patient's prediction result set (i.e., $C_{Y_k} \in S$), the number of people with the disease labeled k is increased by 1. With these statistics, the HP can give some useful suggestion to the relevant institutions, for example, if the number of people with the disease Y_k is higher, then it can suggest the pharmacy to prepare more related drugs. The detailed statistics will be illustrated in Section VI-C2.

4) Prediction Result Retrieving: After receiving the prediction result S^*, for each $C^*_{Y_k} \in S^*$, u_c decrypts it with the secret key K_{u_c} to obtain C_{Y_k}, and then decrypts the C_{Y_k} with the secret key λ to obtain the corresponding disease name Y_k. Note that if S^* is an empty set, it means that u_c may not suffer from any disease. After that, u_c can decide whether he or she needs to consult some specific type of doctors or specialists to follow their recommendations.

V. Privacy Analysis

In this section, we analyze the privacy properties of our EPDP, focusing on how our EPDP can achieve the privacy preservation of MUs and the HP.

A. Privacy-Preservation of MUs

In this section, we discuss the privacy-preservation of confirmed patients and undiagnosed patients, respectively.

1) Privacy-Preservation of Confirmed Patients: In the phase of disease model training, confirmed patients need to provide the historical medical data. Specifically, each confirmed patient u_i $(i = 1, 2, \ldots, l)$ first encrypts $\mathbf{x}^i$, $\mathbf{y}^i$, and $\mathbf{z}^i$ by using the OU encryption algorithm $E(\cdot)$, respectively, and then sends the ciphertexts (C_1^i, C_2^i, C_3^i) to the CP by the secure channel. Because the OU cryptosystem is IND-CPA secure, the CP cannot obtain any plaintext from the obtained ciphertext without the private key of OU cryptosystem.

After l encrypted data have been received, the CP aggregates them and then forwards the aggregated ciphertexts (C_1, C_2, C_3) to the HP. Once these aggregated ciphertexts are received, the HP can decrypt them with the private key to obtain the aggregated data, e.g., $\sum_{i=1}^{l} x_j^i$, $\sum_{i=1}^{l} y_k^i$ and $\sum_{i=1}^{l} x_j^i \cdot y_k^i$ for $j = 1, 2, \ldots, n_s$ and $k = 1, 2, \ldots, n_d$. However, the HP cannot recover the individual data, i.e., x_j^i and y_k^i for $j = 1, 2, \ldots, n_s$ and $k = 1, 2, \ldots, n_d$. The reason mainly contains two aspects: on the one hand, since the communication between confirmed patients and the CP is assumed to be secure, the HP cannot capture the communication information, i.e., (C_1^i, C_2^i, C_3^i) for $i = 1, 2, \ldots, l$. On the other hand, the CP is honest-but-curious, which would strictly follow the scheme, and thus the CP only sends the aggregated ciphertexts (C_1, C_2, C_3) rather than individual ciphertext (C_1^i, C_2^i, C_3^i) to the HP. Therefore, the HP cannot obtain the historical medical data of each confirmed patients even though it has the private key. Note that we do not consider the collusion attack between the CP and HP in this paper.

Similarly, the other patients cannot obtain the privacy of the interested patient because they neither own the private key nor obtain any communication information.

2) Privacy-Preservation of Undiagnosed Patients: In the phase of disease risk prediction, undiagnosed patients want to know whether they may suffer from some diseases by using the service of disease risk prediction. In more details, an undiagnosed patient u_c as a registered member of the HP first generates the query element $H(\lambda, \varphi)$ with the obtained secret key λ. Then, u_c generates the encrypted query C_{u_c} with the CP's public key $\mathrm{PK_{CP}}$ and sends it to the CP for disease diagnosis. Then, the CP can decrypt the C_{u_c} with the private key $\mathrm{SK_{CP}}$ to obtain $H(\lambda, \varphi)$. Naturally, the CP wants to obtain u_c's privacy (i.e., the value φ) from the obtained information $H(\lambda, \varphi)$. However, since $H(\lambda, \cdot)$ is the keyed-cryptographic hash function, the CP cannot get the value φ from $H(\lambda, \varphi)$ without the secret key λ. After executing Algorithm 3, the CP can get the prediction result S containing all encrypted disease names that the patient may suffer. Since each disease name is encrypted by the symmetric encryption algorithm (i.e., AES used in this paper), the CP cannot obtain the plaintext without the symmetric key λ. Therefore, the CP can obtain neither the undiagnosed symptom vector nor the possible suffered diseases for undiagnosed patients.

In addition, other registered patients may also try to know the privacy of the u_c. Specifically, since other registered patients also have the secret key λ, if they obtain $H(\lambda, \varphi)$ or C_{Y_k}, then they can know what disease u_c may suffer from, i.e., Y_k. However, as described in Section IV-C2, u_c encrypts $H(\lambda, \varphi)$ through a secure public-key encryption algorithm $E_{PK_{\mathrm{CP}}}(\cdot)$, e.g., RSA-OAEP. Thus, due to the CCA security of the RSA-OAEP, other registered patients cannot decrypt the ciphertext C_{u_c} to obtain $H(\lambda, \varphi)$ without the private key $\mathrm{SK_{CP}}$. Similarly, since the prediction result is encrypted by the symmetric encryption algorithm, e.g., AES, other registered patients cannot obtain C_{Y_k} without u_c's secret key K_{u_c}. With the same reason, the HP also cannot obtain the sensitive data of interested patients. As a result, our EPDP can protect the privacy of undiagnosed patients from disclosure.

B. Privacy-Preservation of HP

In this section, we discuss how our proposed EPDP can protect the training results, i.e., the probabilities of naïve Bayesian classifier, from disclosure. Specifically, in the phase of disease model training, the aggregated ciphertexts (C_1, C_2, C_3) are encrypted by the OU encryption algorithm, the CP and

MUs cannot get the corresponding plaintexts without the private key (p, q). That is, the CP and MUs cannot obtain the probabilities of naïve Bayesian classifier. Besides, the CP can obtain the BF_k of each disease Y_k from the HP, thus it wants to obtain some useful information, i.e., the symptom vectors that may cause patients to suffer from the disease Y_k. Similar to the above analysis, the CP cannot generate the valid hash value, e.g., $H(\lambda, \mu_i)$, thus it cannot know the corresponding μ_i. In other words, it cannot know which n_s-dimensional binary vector is recognized as the symptom vector included in BF_k. As a result, our EPDP can protect the privacy of the HP from disclosure.

VI. Performance Evaluation

In this section, we analyze the performance of the proposed EPDP in terms of computational cost and communication overhead and make the comparison with the PPCD [27].

A. Computational Cost

In this section, we theoretically analyze the computational cost of our EPDP in terms of two phases, i.e., disease model training and disease risk prediction, and compare it with the PPCD [27].

For the sake of simplicity, we set T_{E_O}, T_{D_O}, and T_{add_O} represent the computational costs of an encryption, a decryption and an additive homomorphic operation in the OU cryptosystem, respectively. We set T_{alg1}, T_{alg2}, T_{alg3}, T_{SE}, and T_H represent computational costs of Algorithms 1–3, AES algorithm and HMAC, respectively. Besides, we use T_{E_R} and T_{D_R} to represent the computational costs of an encryption and a decryption in the RSA-OAEP cryptosystem, respectively. Since the PPCD [27] uses the bilinear pairing technology and Paillier cryptosystem to protect the privacy, for convenience, we use T_e, T_p, T_{et}, and T_{mt} to represent the computational costs of an exponentiation in $\mathbb{G}$, a pairing operation, an exponentiation in $\mathbb{G}_T$ and a multiplication in $\mathbb{G}_T$, respectively. We also use T_{E_P}, T_{D_P}, T_{add_P}, and T_{mul_P} to represent the computational costs of an encryption, a decryption, an additive homomorphic operation and a scalar-multiplicative homomorphic operation in the Paillier cryptosystem, respectively.

1) Computational Cost of Our EPDP: In the phase of disease model training, every confirmed patient generates three ciphertexts by executing the encryption algorithm $E(\cdot)$, which costs $3T_{E_O}$. Thus, the computational costs for l confirmed patients are $3l \cdot T_{E_O}$. Then, the CP aggregates l ciphertexts into three aggregated data, which costs $3(l-1) \cdot T_{\mathrm{add}_O}$. After that, the HP decrypts three aggregated ciphertexts by calling $D(\cdot)$, which spends $3T_{D_O}$. Finally, the HP can obtain the aggregated data by executing Algorithm 1, which costs $3 \cdot T_{\mathrm{alg1}}$. After obtaining the probabilities of naïve Bayesian classifier, for each disease Y_k, the HP generates the corresponding BF_k by calling Algorithm 2. Thus, the costs for n_d diseases are $n_d \cdot T_{\mathrm{alg2}}$. Besides, in order to protect the privacy, the HP encrypts each disease's name through AES algorithm before outsourcing. The corresponding costs are $n_d \cdot T_{\mathrm{SE}}$.

In the phase of disease risk prediction, a registered undiagnosed patient first generates the encrypted query C_{u_c}, which costs $T_H + T_{E_R}$. Then, the CP decrypts C_{u_c} to obtain $H(\lambda, \varphi)$, and obtains the prediction result by calling Algorithm 3. The corresponding costs are $T_{D_R} + T_{\mathrm{alg3}}$. After obtaining the prediction result S, for each $C_{Y_k} \in S$, the CP computes $C^*_{Y_k} = \mathrm{SE}(K_{u_c}, C_{Y_k})$. Thus, the total costs for $|S|$ elements are $|S| \cdot T_{\mathrm{SE}}$. After obtaining the result S^*, for each $C^*_{Y_k} \in S^*$, this patient decrypts it with K_{u_c} to obtain C_{Y_k}, and then gets the disease name Y_k with λ by performing the AES decryption. Thus, the corresponding costs are $2|S| \cdot T_{\mathrm{SE}}$.

2) Computational Cost of the PPCD [27]: In order to make a comparison, we briefly describe the computational costs of the PPCD [27]. In the phase of disease model training, l confirmed patients first encrypt the historical data and send them to the CP for aggregation, which cost $l(n_s + n_d)(T_e + T_p + 3T_{mt}) + (2l+1)(n_s + n_d)T_{\mathrm{et}}$. It is worth noting that the HP uses the naïve Bayesian classifier to achieve the disease diagnosis. Based on the definition of the naïve Bayesian classifier [23], [37], the conditional probability $P(X_j = 1|Y_k = 1)$ is computed as

$$P\big(X_j = 1|Y_k = 1\big) = \frac{\sum_{i=1}^{l} x_j^i y_k^i}{\sum_{i=1}^{l} y_k^i}. \tag{13}$$

However, in the PPCD [27], the conditional probability is computed as $P(X_j = 1|Y_k = 1) = \sum_{i=1}^{l} x_j^i / \sum_{i=1}^{l} y_k^i$, which is not precise. The reason is that when $x_j^i = 1$, y_k^i may be 0 or 1. Thus, it is not precise for computing $P(X_j = 1|Y_k = 1)$ to directly use $\sum_{i=1}^{l} x_j^i$ to replace the $\sum_{i=1}^{l} x_j^i y_k^i$. Based on (13), every confirmed patient should also encrypt $n_s n_d$ data, i.e., $E(x_j^i y_k^i)$, for $j = 1, 2, \ldots, n_s$, $k = 1, 2, \ldots, n_d$. Hence, the real computational costs of the disease model training are $l(n_s + n_d + n_s n_d)(T_e + T_p + 3T_{mt}) + (2l+1)(n_s + n_d + n_s n_d)T_{\mathrm{et}}$. In the phase of disease risk prediction, the Paillier cryptosystem [29] has been used to realize the privacy-preserving disease diagnosis. First, it requires $((8n_s - 6)n_d + n_s)T_{E_P} + (4n_s - 4)n_d \cdot T_{D_P} + (14n_s - 10)n_d \cdot T_{\mathrm{add}_P} + (12n_s - 4)n_d \cdot T_{\mathrm{mul}_P}$ to calculate the disease risk of an undiagnosed patient. After that, it costs $6n_d \cdot T_{E_P} + n_d \cdot T_{D_P} + 11n_d \cdot T_{\mathrm{add}_P} + 8n_d \cdot T_{\mathrm{mul}_P}$ to judge whether this patient suffers from some specific diseases.

3) Comparison: We present the comparison of computational cost for our EPDP and the PPCD [27] in Table II. From the table, we can see that the computational costs of our EPDP are less than that of the PPCD, which will be further shown in our simulations in Section VI-C.

B. Communication Overhead

In this section, we theoretically analyze the communication overhead of our EPDP, and then make a comparison with the PPCD [27].

1) Communication Overhead of Our EPDP: In the phase of disease model training, each confirmed patient sends (C_1^i, C_2^i, C_3^i) to the CP. Since the security parameter of OU cryptosystem is κ_0 (see Section IV-A), the bit length of each generated ciphertext is $3\kappa_0$. That is, each confirmed patient spends $9\kappa_0$ bits in length to transmit the encrypted data. Thus, the overheads for l confirmed patients are $9l \cdot \kappa_0$ bits. After receiving l encrypted data, the CP aggregates them and forwards the aggregated ciphertexts (C_1, C_2, C_3) to the HP,

TABLE II
THEORETICAL COMPARISON OF COMPUTATIONAL COST

Scheme	Disease model training	Disease risk prediction
Our EPDP	$3l \cdot T_{E_O} + 3(l-1)T_{add_O} + 3(T_{D_O} + T_{alg1}) + n_d(T_{alg2} + T_{SE})$	$T_H + T_{E_R} + T_{D_R} + T_{alg3} + 3\|S\| \cdot T_{SE}$
PPCD [27]	$ln \cdot (T_e + T_p) + 3ln \cdot T_{mt} + (2l+1)n \cdot T_{et}$	$(8n_d+1)n_s \cdot T_{E_P} + (4n_s-3)n_d \cdot T_{D_P} +$ $(14n_s+1)n_d \cdot T_{add_P} + (12n_s+4)n_d \cdot T_{mul_P}$

$n = n_s + n_d + n_s n_d$.

TABLE III
THEORETICAL COMPARISON OF COMMUNICATION OVERHEAD (BITS)

Scheme	Disease model training	Disease risk prediction
Our EPDP	$(9l+9)\kappa_0$	$\|S\| \cdot \|Y_k\| + 2\kappa_0$
PPCD [27]	$n\,(l\|\mathbb{G}\| + (l+2)\|\mathbb{G}_T\|)$	$(32n_s n_d + 8n_s + 28n_d)\kappa_0$

$n = n_s + n_d + n_s n_d$.

which costs $9\kappa_0$ bits to transmit. In the phase of disease risk prediction, an undiagnosed patient sends the encrypted query C_{u_c} to the CP, which costs $2\kappa_0$ bits if we set the bit length of the security parameter in RSA to the same as the OU cryptosystem. After completing the disease risk diagnosis, the CP sends the prediction result S^* to this patient. Note that the elements in S^* are ciphertexts generated by the AES algorithm, thus the corresponding overheads are $|S| \cdot |Y_k|$, where $|S|$ is the number of the elements in the set S^* (i.e., S) and $|Y_k|$ is the bit length of disease name Y_k.

2) Communication Overhead of the PPCD [27]: In the phase of disease model training, the PPCD uses the bilinear pairing technique to protect the privacy. In order to facilitate expression, we set $|\mathbb{G}|$ and $|\mathbb{G}_T|$ denote the bit length of the element in $\mathbb{G}$ and $\mathbb{G}_T$, respectively. Specifically, each confirmed patient spends $(n_s + n_d + n_s n_d) \cdot (|\mathbb{G}| + |\mathbb{G}_T|)$ bits to transmit his or her encrypted historical medical data to the CP under the correct calculations of the naïve Bayesian classifier. Accordingly, the corresponding overheads for l confirmed patients are $l(n_s + n_d + n_s n_d) \cdot (|\mathbb{G}| + |\mathbb{G}_T|)$ bits. Then, the CP aggregates l data and sends them to the HP, which costs $(n_s + n_d + n_s n_d) \cdot 2|\mathbb{G}_T|$ bits to transmit. In the phase of disease risk prediction, an undiagnosed patient sends n_s ciphertexts encrypted by Paillier cryptosystem to the HP via the CP, which costs $2n_s \cdot 4\kappa_0$ bits if we set the security parameter of Paillier cryptosystem to κ_0. Then, the HP needs to interact with this patient to compute the disease risk, which costs $(32n_s - 16)n_d \cdot \kappa_0$ bits. After that, the HP outsources n_d results to the CP for prediction, which costs $16n_d \cdot \kappa_0$ bits. Once receiving n_d results, the CP interacts with this patient to make the judgement, which costs $28n_d \cdot \kappa_0$ bits.

3) Comparison: We give the comparison of communication overhead for our EPDP and the PPCD [27] in Table III. Note that κ_0 is the security parameter of OU and Paillier cryptosystems, κ_2 is the bit length of the hash value, $|S|$ is the number of the elements in set S such that $|S| \leq n_d$ and $|\mathbb{G}|$ ($|\mathbb{G}_T|$) is the bit length of the element in the group $\mathbb{G}$ ($\mathbb{G}_T$). From the table, we can see the communication overhead of our EPDP is much less than that of the PPCD. The details will be described in Section VI-C.

C. Simulation

In this section, we conduct the experiments in Java running on the MacBook Pro with one 2.3-GHz Intel Core i5 and 8-GB memory. Similar to [27], we consider two datasets. One real dataset is used from the UCI machine learning repository called acute inflammations dataset (AID) [38]. We use the AID to test the accuracy of the prediction for our EPDP and the performance of our EPDP and the PPCD [27] in terms of computational and communication overheads. We also use the synthetic dataset to test all factors affecting the performance of both schemes.

1) Simulation Setup: In the simulation, we set the security parameter κ_0 for both OU and Paillier cryptosystems as $\kappa_0 = 512$. We apply the RSA-OAEP for the secure public-key cryptosystem, where the RSA modulus is set to 1024 bits. We choose AES-256 and HMAC-SHA1 as the symmetric encryption algorithm $\mathrm{SE}(\lambda, \cdot)$ and keyed cryptographic hash function $H(\lambda, \cdot)$, respectively, i.e., $\kappa_1 = 256$ and $\kappa_2 = 160$. For the bilinear pairing parameters in the PPCD, we choose type A elliptic curve with 512-bits base field size [39]. Based on the analysis for (3), we set $f = 88$ and $L = 2^{27}$ so that the BF can contain up to 2^{20} elements and the corresponding false positive probability is about 10^{-27}. Besides, the details of two datasets are described as follows.

a) *Real Dataset (AID):* The AID was created by a medical expert as a dataset to test the expert system, which contains 120 instances used to perform the presumptive diagnosis of two diseases of the urinary system. Each instance contains six symptom attributes and two diseases [inflammation of urinary bladder (IUB) and nephritis of renal pelvis origin (NRPU)], i.e., $n_s = 6$ and $n_d = 2$. We use first 80 instances for disease model training and the remaining 40 instances for disease risk prediction.

b) *Synthetic Dataset:* In order to test all the factors that affect both schemes, we use the synthetic dataset to test. The randomly generated synthetic dataset consists of 500 tuples with 10 attributes. The value of each element is randomly picked either 0 or 1. There are three factors which affect the total running time of both schemes: a) the number of historical medical data (l); b) the number of symptom attributes (n_s); and c) the number of diseases categories (n_d).

2) Simulation Results: The accuracy of the disease risk prediction for our EPDP over the AID is shown in Table IV. From the table, we can see that 40 instances used for prediction can be successfully classified. Besides, we test the efficiency about our EPDP and the PPCD [27] in terms of computational costs and communication overhead, which is given in Table V.

TABLE IV
ACCURACY OF DISEASE RISK PREDICTION FOR OUR EPDP OVER THE AID

Disease name	IUB	NRPU
1	14/14(100%)	32/32(100%)
0	26/26(100%)	8/8(100%)

1: getting the disease; 0: not getting disease.

TABLE V
EFFICIENCY COMPARISON OVER THE AID

Scheme	Computational cost (s)		Communication overhead (KB)	
	Training	Prediction	Training	Prediction
Our EPDP	0.84	0.17	45.56	6.43
PPCD [27]	33.39	1.06	305	1220

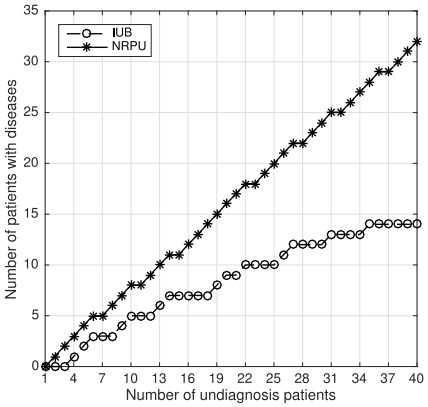


Fig. 3. Statistics of disease prediction.

From the table, we can see that the efficiency of our EPDP is much better than that of the PPCD, especially in the phase of disease risk prediction. As described in Section IV-C3, in addition to completing the disease diagnosis on behalf of the HP, the CP can also help the HP to collect statistical information based on the prediction results. In more details, Fig. 3 shows the statistical information of the predicted prevalence among undiagnosed patients. From the figure, we can see that the number of patients suffering from NRPU is higher than the number of patients suffering from IUB. After obtaining this result, the HP can give some useful suggestion to the relevant institutions, for example, it can suggest the pharmacy to prepare more drugs for the NRPU.

In the simulation of the synthetic dataset, we consider the computational cost and communication overhead for our EPDP and the PPCD [27] varying with the number of symptom attributes n_s, the number of disease categories n_d, and the number of historical medical data l. Specifically, we depict the comparison of computational cost for our EPDP and the PPCD [27] in Fig. 4. It is shown that the computational costs of our EPDP are much less than that of the PPCD. The reason is that in the phase of disease model training, we use a super-increasing sequence vector $\mathbf{a}$ to compress $(n_s + n_d + n_s n_d)$ encrypted operations into three, which also largely reduces related computational costs, e.g., aggregation and decryption operations. Besides, our EPDP uses the BF technique to predict the disease risk, as described in Algorithm 3, which only needs to perform the efficient membership query operation. However, the PPCD executes a great number of secure multiplication (SM) protocol [40], which needs to pay a high computational price. Note that only the overhead of disease model training is influenced by l, so when l changes, we only plot the computational cost of training phase in Fig. 4(c).

In Fig. 5, we plot the comparison of communication overhead for our EPDP and the PPCD [27] in terms of n_s, n_d, and l. It is shown that the communication overheads for our EPDP are much less than the PPCD. Specifically speaking, in the phase of disease model training, each confirmed patient only needs to send three compressed ciphertexts in our EPDP without being affected by n_s and n_d, but the PPCD requires each confirmed patient to transmit $(n_s n_d + n_s + n_d)$ ciphertexts. In the phase of disease risk prediction, our EPDP only needs the CP to perform the membership query operation $\text{QueryBF}_k(\cdot)$ to complete the disease prediction, which does not need any interaction operations compared with the PPCD.

Based on the above analyses, our EPDP can achieve efficient and acceptable computing and communication, which is more suitable for real-time e-healthcare environment than the PPCD [27].

VII. RELATED WORKS

Disease risk prediction has been widely investigated [41]–[43], since it can significantly facilitate patients and HPs, for example, empower patients to manage their own health, reduce office visits to get results, and improve decisions, etc. However, these works do not consider the privacy-preserving issue, which is a necessary factor in the disease risk prediction research [44], [45]. Therefore, it is preferred to design a privacy-preserving disease risk prediction scheme.

Recently, many privacy-preserving disease risk prediction schemes have been proposed. For example, Wang *et al.* [46] presented a feasible privacy-preserving single-layer perceptron scheme to obtain the disease model. Vaidya *et al.* first suggested the privacy-preserving naïve Bayesian classifier in [47]. Rahulamathavan *et al.* [22] proposed a privacy-preserving clinical decision support system using a Gaussian kernel-based support vector machine. However, these schemes only deal with the phase of disease model training. In order to achieve the privacy-preserving disease prediction, some similarity matching or statistic analysis technologies have been conducted. For instance, Wang *et al.* [28] advised the smartphone-based preclinical guidance scheme to provide the disease risk diagnosis. Zhou *et al.* [4] gave a secure and efficient privacy-preserving dynamic medical text mining and image feature extraction scheme. Shemeikka *et al.* [48] discussed computer-based decision-support systems to assist intensive care unit physicians to manage the infectious diseases. Nevertheless, these schemes did not take the disease model training into consideration. Obviously, the aforementioned works achieved either the privacy-preserving disease model training or the privacy-preserving disease prediction, but not both parts. The direct combination of the above schemes seems difficult to apply in the disease risk prediction,

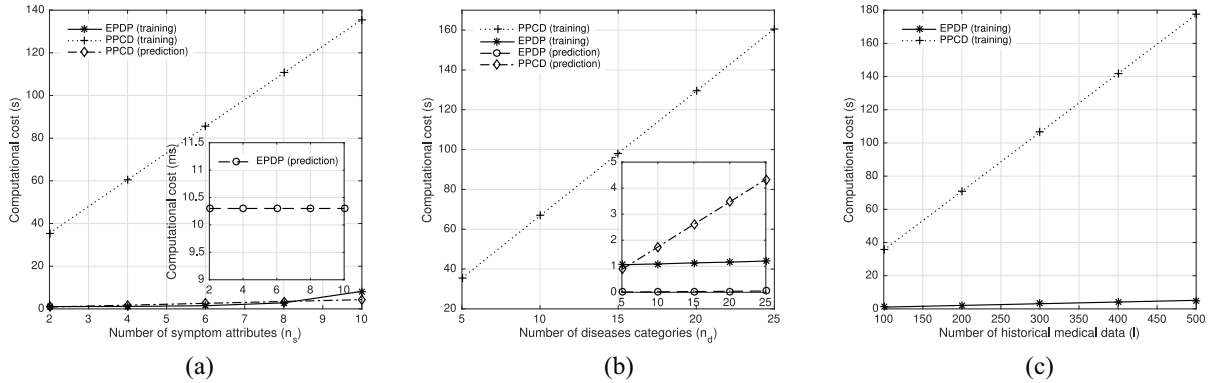


Fig. 4. Comparison of computational cost between our EPDP and the PPCD [27]. (a) $n_d = 5$ and $l = 100$. (b) $n_s = 2$ and $l = 100$. (c) $n_s = 2$ and $n_d = 5$.

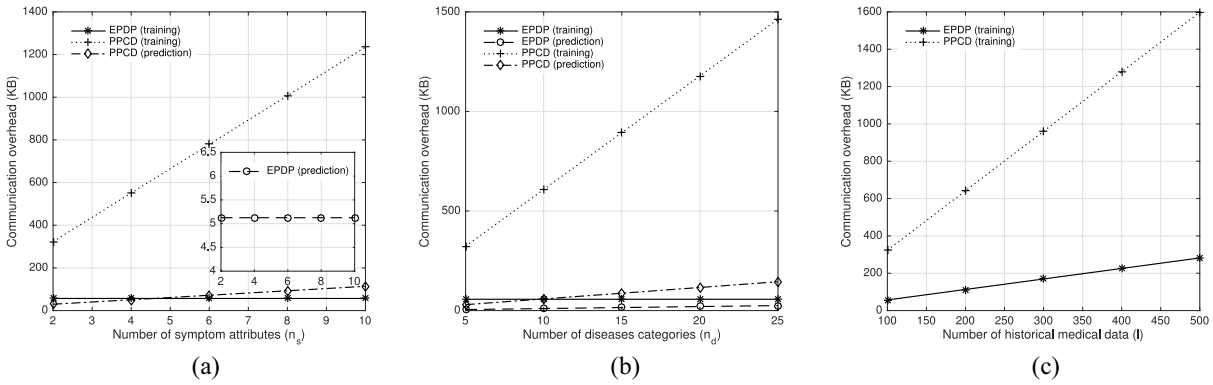


Fig. 5. Comparison of communication overhead between our EPDP and the PPCD [27]. (a) $n_d = 5$ and $l = 100$. (b) $n_s = 2$ and $l = 100$. (c) $n_s = 2$ and $n_d = 5$.

because the prediction methods corresponding to different training methods are also different, let alone operate in the ciphertext domain. That is, these works cannot achieve the comprehensiveness described in Section I. To address this disadvantage, Liu *et al.* [27] presented a new privacy-preserving patient-centric clinical decision support system. This scheme utilizes the naïve Bayesian classifier [23], [37] to complete the disease model training and the SM protocol [40] to predict disease. However, owing to the time-consuming pairing technology [49] and Paillier encryption algorithm [29]–[32], the computational and communication overheads are relatively high, which is not suitable for the real-time e-healthcare, especially medical emergency. Conceivably, existing schemes do not achieve comprehensiveness, efficiency and privacy-preservation at the same time. This gap motivates our work in this paper.

VIII. Conclusion

In this paper, we have proposed an efficient and privacy-preserving disease risk prediction scheme in e-healthcare, named EPDP. First, the OU cryptosystem has been used to protect the privacy, which serves as the basis of our EPDP. Then, a super-increasing sequence has been introduced to reduce the computational and communication overheads, and the symptom vector set of each disease is extracted by taking advantage of naïve Bayesian classifier in the phase of disease model training. Finally, based on the extracted symptom vector sets, the disease prediction result is obtained by using the efficient BF technique. Detailed privacy analysis shows that our EPDP really achieves the privacy requirements of MUs and the HP in the honest-but-curious model. Furthermore, extensive simulations demonstrate that our EPDP is much more efficient than the existing competing scheme in terms of the computational and communication overheads, and hence our EPDP is more suitable for the real-time e-healthcare environment, especially medical emergency.

Future research includes achieving the message integrity in e-healthcare since it directly influences the accuracy of diagnosis results and even the life safety of patients. Besides, MUs may be unfamiliar with the diagnosed diseases due to lack of the professional knowledge. Therefore, they can authorize some medical professionals to access the prediction results for further guidance of medical care. That is, how to achieve the access control will also be our future work.

References

[1] X. Li *et al.*, "Smart community: An Internet of Things application," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 68–75, Nov. 2011.

[2] G. Eysenbach, "What is e-health?" *J. Med. Internet Res.*, vol. 3, no. 2, p. e20, 2001.

[3] J. Sun, Y. Fang, and X. Zhu, "Privacy and emergency response in e-healthcare leveraging wireless body sensor networks," *IEEE Wireless Commun.*, vol. 17, no. 1, pp. 66–73, Feb. 2010.

[4] J. Zhou, Z. Cao, X. Dong, and X. Lin, "PPDM: A privacy-preserving protocol for cloud-assisted e-healthcare systems," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 7, pp. 1332–1344, Oct. 2015.

[5] C. A. Schurink, P. J. Lucas, I. M. Hoepelman, and M. J. Bonten, "Computer-assisted decision support for the diagnosis and treatment of infectious diseases in intensive care units," *Elsevier Lancet Infectious Diseases*, vol. 5, no. 5, pp. 305–312, 2005.

[6] K.-P. Lin and M.-S. Chen, "On the design and analysis of the privacy-preserving SVM classifier," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 11, pp. 1704–1717, Nov. 2011.

[7] H. Yu, X. Jiang, and J. Vaidya, "Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data," in *Proc. ACM Symp. Appl. Comput. (SAC)*, Dijon, France, Apr. 2006, pp. 603–610.

[8] A. Toninelli, R. Montanari, and A. Corradi, "Enabling secure service discovery in mobile healthcare enterprise networks," *IEEE Wireless Commun.*, vol. 16, no. 3, pp. 24–32, Jun. 2009.

[9] Y. Ren, R. Werner, N. Pazzi, and A. Boukerche, "Monitoring patients via a secure and mobile healthcare system," *IEEE Wireless Commun.*, vol. 17, no. 1, pp. 59–65, Feb. 2010.

[10] M. Li, W. Lou, and K. Ren, "Data security and privacy in wireless body area networks," *IEEE Wireless Commun.*, vol. 17, no. 1, pp. 51–58, Feb. 2010.

[11] J. Zhou, Z. Cao, X. Dong, X. Lin, and A. V. Vasilakos, "Securing m-healthcare social networks: Challenges, countermeasures and future directions," *IEEE Wireless Commun.*, vol. 20, no. 4, pp. 12–21, Aug. 2013.

[12] C. Zuo, J. Shao, J. K. Liu, G. Wei, and Y. Ling, "Fine-grained two-factor protection mechanism for data sharing in cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 186–196, Jan. 2018.

[13] C. Zuo, J. Shao, G. Wei, M. Xie, and M. Ji, "CCA-secure ABE with outsourced decryption for fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 730–738, Jan. 2018.

[14] H. Mohammadhassanzadeh, W. V. Woensel, S. R. Abidi, and S. S. R. Abidi, "Semantics-based plausible reasoning to extend the knowledge coverage of medical knowledge bases for improved clinical decision support," *BioData Min.*, vol. 10, no. 1, pp. 1–7, 2017.

[15] K. Zhang, X. Liang, J. Ni, K. Yang, and X. S. Shen, "Exploiting social network to enhance human-to-human infection analysis without privacy leakage," *IEEE Trans. Depend. Secure Comput.*, vol. 5, no. 4, pp. 607–620, Jul./Aug. 2018.

[16] K. Zhang *et al.*, "Security and privacy for mobile healthcare networks: From a quality of protection perspective," *IEEE Wireless Commun.*, vol. 22, no. 4, pp. 104–112, Aug. 2015.

[17] K. Zhang and X. S. Shen, *Security and Privacy for Mobile Healthcare Networks* (Wireless Networks). Cham, Switzerland: Springer, 2015.

[18] 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.

[19] R. Lu, X. Lin, and X. S. Shen, "SPOC: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 3, pp. 614–624, Mar. 2013.

[20] R. Lafta *et al.*, "A fast Fourier transform-coupled machine learning-based ensemble model for disease risk prediction using a real-life dataset," in *Proc. 21st Pac.–Asia Conf. Adv. Knowl. Disc. Data Min. (PAKDD)*, May 2017, pp. 654–670.

[21] B. K. Mishra, S. Mishra, S. Sahoo, and B. Panda, "Impact of swarm intelligence techniques in diabetes disease risk prediction," *Int. J. Knowl. Disc. Bioinformat.*, vol. 6, no. 2, pp. 29–43, 2016.

[22] Y. Rahulamathavan, S. Veluru, R. C.-W. Phan, J. A. Chambers, and M. Rajarajan, "Privacy-preserving clinical decision support system using Gaussian kernel-based classification," *IEEE J. Biomed. Health Inform.*, vol. 18, no. 1, pp. 56–66, Jan. 2014.

[23] K.-L. Liu and T.-T. Wong, "Naïve Bayesian classifiers with multinomial models for rRNA taxonomic assignment," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 10, no. 5, p. 1, Sep./Oct. 2013.

[24] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[25] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Adv. Cryptol. (EUROCRYPT)*, Espoo, Finland, May/Jun. 1998, pp. 308–318.

[26] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1621–1631, Sep. 2012.

[27] X. Liu, R. Lu, J. Ma, L. Chen, and B. Qin, "Privacy-preserving patient-centric clinical decision support system on Naïve Bayesian classification," *IEEE J. Biomed. Health Inform.*, vol. 20, no. 2, pp. 655–668, Mar. 2016.

[28] G. Wang, R. Lu, and C. Huang, "Pguide: An efficient and privacy-preserving smartphone-based pre-clinical guidance scheme," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

[29] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Adv. Cryptol. (EUROCRYPT)*, Prague, Czech Republic, May 1999, pp. 223–238.

[30] Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk, "Generating private recommendations efficiently using homomorphic encryption and data packing," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1053–1066, Jun. 2012.

[31] V. Nikolaenko *et al.*, "Privacy-preserving ridge regression on hundreds of millions of records," in *Proc. IEEE Symp. Security Privacy (SP)*, Berkeley, CA, USA, May 2013, pp. 334–348.

[32] Y. Zheng, H. Duan, and C. Wang, "Learning the truth privately and confidently: Encrypted confidence-aware truth discovery in mobile crowdsensing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2475–2489, Oct. 2018.

[33] A. Z. Broder and M. Mitzenmacher, "Survey: Network applications of bloom filters: A survey," *Internet Math.*, vol. 1, no. 4, pp. 485–509, 2003.

[34] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," Internet Eng. Task Force, Fremont, CA, USA, RFC 2104, pp. 1–11, 1997.

[35] F. V. Jensen, *An Introduction to Bayesian Networks*, vol. 210. London, U.K.: UCL Press, 1996.

[36] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern, "RSA-OAEP is secure under the RSA assumption," in *Proc. 21st Annu. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO)*, Santa Barbara, CA, USA, Aug. 2001, pp. 260–274.

[37] C. Hsu, H. Huang, and T. Wong, "Why discretization works for naive Bayesian classifiers," in *Proc. 17th Int. Conf. Mach. Learn. (ICML)*, Jun./Jul. 2000, pp. 399–406.

[38] (2009). *Acute Inflammations Data Set, UCI Machine Learning Repository*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets.html

[39] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2011, pp. 850–855. [Online]. Available: http://gas.dia.unisa.it/projects/jpbc/

[40] B. K. Samanthula, Y. Elmehdwi, and W. Jiang, "*k*-Nearest neighbor classification over semantically secure encrypted relational data," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1261–1273, May 2015.

[41] M. Caron, R. Allard, L. Bédard, J. Latreille, and D. L. Buckeridge, "Enteric disease episodes and the risk of acquiring a future sexually transmitted infection: A prediction model in montreal residents," *J. Amer. Med. Informat. Assoc.*, vol. 23, no. 6, pp. 1159–1165, 2016.

[42] Y. Shen *et al.*, "Risk prediction for cardiovascular disease using ECG data in the China Kadoorie Biobank," in *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Orlando, FL, USA, Aug. 2016, pp. 2419–2422.

[43] M. Nagata, K. Matsumoto, and M. Hashimoto, "Prediction for disease risk and medical cost using time series healthcare data," in *Proc. 9th Int. Joint Conf. Biomed. Eng. Syst. Technol. (BIOSTEC)*, vol. 5, Feb. 2016, pp. 517–522.

[44] S. Jiang, X. Zhu, and L. Wang, "EPPS: Efficient and privacy-preserving personal health information sharing in mobile healthcare social networks," *Sensors*, vol. 15, no. 9, pp. 22419–22438, 2015.

[45] J. Zhou, X. Lin, X. Dong, and Z. Cao, "PSMPA: Patient self-controllable and multi-level privacy-preserving cooperative authentication in distributedm-healthcare cloud computing system," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 6, pp. 1693–1703, Jun. 2015.

[46] G. Wang, R. Lu, and C. Huang, "PSLP: Privacy-preserving single-layer perceptron learning for e-healthcare," in *Proc. 10th Int. Conf. Inf. Commun. Signal Process. (ICICS)*, Singapore, Dec. 2015, pp. 1–5.

[47] J. Vaidya, M. Kantarcioglu, and C. Clifton, "Privacy-preserving naïve Bayes classification," *VLDB J.*, vol. 17, no. 4, pp. 879–898, 2008.

[48] T. Shemeikka *et al.*, "A health record integrated clinical decision support system to support prescriptions of pharmaceutical drugs in patients with reduced renal function: Design, development and proof of concept," *Int. J. Med. Informat.*, vol. 84, no. 6, pp. 387–395, 2015.

[49] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. 21st Annu. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO)*, Santa Barbara, CA, USA, Aug. 2001, pp. 213–229.

Xue Yang received the B.S. degree in information security from Southwest Jiaotong University, Chengdu, China, in 2012, where she is currently pursuing the Ph.D. degree in information and communication engineering.

Her current research interests include big data security and privacy, applied cryptography, and network security.

Rongxing Lu (S'99–M'11–SM'15) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2012.

He was an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from 2013 to 2016. He has been an Assistant Professor with the Faculty of Computer Science (FCS), University of New Brunswick (UNB), Fredericton, NB, Canada, since 2016. He was a Post-Doctoral Fellow with the University of Waterloo, from 2012 to 2013. His current research interests include applied cryptography, privacy enhancing technologies, and IoT-big data security and privacy. He has been published extensively in the above areas.

Dr. Lu was a recipient of the most prestigious Governor Generals Gold Medal, the Eighth IEEE Communications Society (ComSoc) Asia–Pacific Outstanding Young Researcher Award, in 2013, and eight Best Student Paper Awards from some reputable journals and conferences. He was the recipient of the 2016–2017 Excellence in Teaching Award, FCS, UNB. He currently serves as the Vice-Chair (Publication) of the IEEE ComSoc Communications and Information Security Technical Committee. He is currently a Senior Member of the IEEE Communications Society.

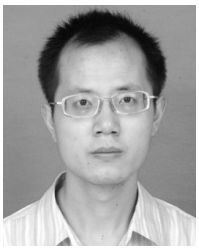
Jun Shao received the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2008.

He was a Post-Doctoral Fellow with the School of Information Sciences and Technology, Pennsylvania State University, Pennsylvania, PA, USA, from 2008 to 2010. He is currently a Professor with the School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, China. His current research interests include network security and applied cryptography.

Xiaohu Tang (M'04) received the B.S. degree in applied mathematics from Northwest Polytechnic University, Xi'an, China, in 1992, the M.S. degree in applied mathematics from Sichuan University, Chengdu, China, in 1995, and the Ph.D. degree in electronic engineering from Southwest Jiaotong University, Chengdu, in 2001.

From 2003 to 2004, he was a Research Associate with the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Hong Kong. From 2007 to 2008, he was a Visiting Professor with the University of Ulm, Ulm, Germany. Since 2001, he has been with the School of Information Science and Technology, Southwest Jiaotong University, where he is currently a Professor. His current research interests include coding theory, network security, distributed storage, and information processing for big data.

Dr. Tang was a recipient of the National Excellent Doctoral Dissertation Award in 2003 in China, the Humboldt Research Fellowship in 2007 in Germany, and the Outstanding Young Scientist Award by NSFC in 2013 in China. He serves as an Associate Editors for several journals including the IEEE TRANSACTIONS ON INFORMATION THEORY and *IEICE Transactions on Fundamentals* and has served on a number of Technical Program Committees of conferences.



Haomiao Yang (S'13–M'18) received the M.S. and Ph.D. degrees in computer applied technology from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2004 and 2008, respectively.

He was a Post-Doctoral Fellow with the Research Center of Information Cross over Security, Kyungil University, Gyeongsan, South Korea, from 2012 to 2013. He is currently an Associate Professor with the School of Computer Science and Engineering and Center for Cyber Security, UESTC. His current research interests include cryptography, cloud security, and the cyber security for aviation communication.