Efficient and Privacy-preserving Fog-assisted Health Data Sharing Scheme

WENJUAN TANG and JU REN, Central South University, China
KUAN ZHANG, University of Nebraska–Lincoln, U.S.
DEYU ZHANG and YAOXUE ZHANG, Central South University, China
XUEMIN (SHERMAN) SHEN, University of Waterloo, Canada

Pervasive data collected from e-healthcare devices possess significant medical value through data sharing with professional healthcare service providers. However, health data sharing poses several security issues, such as access control and privacy leakage, as well as faces critical challenges to obtain efficient data analysis and services. In this article, we propose an efficient and privacy-preserving fog-assisted health data sharing (PFHDS) scheme for e-healthcare systems. Specifically, we integrate the fog node to classify the shared data into different categories according to disease risks for efficient health data analysis. Meanwhile, we design an enhanced attribute-based encryption method through combination of a personal access policy on patients and a professional access policy on the fog node for effective medical service provision. Furthermore, we achieve significant encryption consumption reduction for patients by offloading a portion of the computation and storage burden from patients to the fog node. Security discussions show that PFHDS realizes data confidentiality and fine-grained access control with collusion resistance. Performance evaluations demonstrate cost-efficient encryption computation, storage and energy consumption.

CCS Concepts: • **Security and privacy** → **Access control**; • **Human-centered computing** → *Ubiquitous computing*; • **Applied computing** → *Health care information systems*;

Additional Key Words and Phrases: Fog computing, access control, data sharing, e-healthcare, privacy-preservation

ACM Reference format:
Wenjuan Tang, Ju Ren, Kuan Zhang, Deyu Zhang, Yaoxue Zhang, and Xuemin (Sherman) Shen. 2019. Efficient and Privacy-preserving Fog-assisted Health Data Sharing Scheme. *ACM Trans. Intell. Syst. Technol.* 10, 6, Article 68 (October 2019), 23 pages.
https://doi.org/10.1145/3341104

This work is supported by the National Natural Science Foundation of China (No. 61702562 and 61702561), 111 project (No. B18059), International Science & Technology Cooperation Program of China (No. 2013DFB10070), and China Hunan Provincial Science & Technology Program (No. 2012GK4106).

Authors' addresses: W. Tang, J. Ren (corresponding author), D. Zhang, and Y. Zhang, School of Compmuter Science and Engineering, Central South University, Changsha 410083, China; emails: {wenjuantang, renju, zdy876, zyx}@csu.edu.cn; K. Zhang, Department of Electrical and Computer Engineering, University of Nebraska–Lincoln, Lincoln, NE 68588; email: kuan.zhang@unl.edu; X. (Sherman) Shen, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, N2L 3G1, Canada; email: sshen@uwaterloo.ca.

2157-6904/2019/10-ART68 $15.00
https://doi.org/10.1145/3341104

1 INTRODUCTION

The e-healthcare system has emerged as a promising healthcare paradigm for real-time health data collection and health monitoring with the development of information and communication technologies. Various categories of health data collected from heterogenous healthcare devices (e.g., smart health watch and blood glucose meter) may reach about 12 ZBs by 2020 [18], bringing a critical big data management issue. Since cloud computing can store and manage huge volumes of data with its powerful storage resources and computing services, cloud-assisted e-healthcare systems [45] (CAEHS) have been widely developed. In CAEHS, patients use pervasive healthcare devices to collect health data, and then store these data on a remote cloud server, to share with data users. Professional healthcare service providers access and analyze the shared data to provide healthcare services, such as infection analysis, personal treatments, and clinical diagnosis [41].

Privacy leakage and security threats may occur during data sharing in cloud-assisted e-healthcare systems [10]. First, personal privacy might be disclosed during data sharing. For example, employees in the third-party cloud company might obtain the health information and even trade it for money in an illegal market. Second, unauthorized users might access the shared data collected from patients. For example, some unscrupulous pharmaceutical companies might analyze the health data and obtain patients' health status to spread advertisements and drug promotions. Third, the shared health data may be tampered with during data transmission from data collection to storage [28]. For instance, the blood glucose of patients may be altered when it is delivered to healthcare centers, leading to incorrect healthcare treatments [13]. To protect the shared data against privacy leakage, unauthorized data access, and data tampering, Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [33, 42] is widely proposed for health data sharing, because it can support multiple data accessing paradigms with data confidentiality preservation. Patients define access policies to encrypt their shared data and send the ciphertext to the cloud server. Data users access the shared data and decrypt the ciphertext only if their attributes satisfy access policies.

However, existing data sharing schemes [31] still face several challenges in e-healthcare systems. Specifically, they can hardly enable effective healthcare service provision and efficient medical analysis, as well as incur severe resource consumption for encryption in resource-limited e-healthcare devices. First, existing access policy definition in health data sharing schemes cannot guarantee effective healthcare service provision with privacy preservation simultaneously [23]. Patients may generally have their coarse preferences to define access policies according to their experiences and interests. Nonetheless, considering the health data requires professional medical knowledge to understand, patients without sufficient healthcare background can hardly define an appropriate access policy to guarantee privacy preservation and obtain healthcare services simultaneously. Unfortunately, privacy preservation and appropriate healthcare services contradict with each other. If the access policy is defined with "strict" attributes for privacy preservation, the shared data may not be accessible by appropriate healthcare service providers. If the access policy is defined with "loose" attributes, then the shared data can be accessible with more healthcare service providers but may increase the privacy disclosure risks [36]. Here, "strict" access policy means that the defined access policy requires relatively more attributes to satisfy than the "loose" access policy. For example, Alice intends to share her health data for healthcare services with an access policy of $\{top - three\ hospital,\ more\ than\ 20\ years'\ working\ experience\}$. If Alice suffers from cardiovascular disease risks, then cardiovascular professionals can be effective healthcare service providers. However, other non-cardiovascular service providers satisfy the access policy can also access her shared data, which may incur potential privacy leakage. Meanwhile, this kind of health data sharing schemes impair service providers to efficiently analyze various kinds of raw data collected from heterogenous devices. Different from the clinical data that is highly targeted to the specific disease diagnosis [17], the shared data of e-healthcare systems are aggregated and mixed from different categories [46]. Service providers can hardly seek out the specific kind of healthcare data for corresponding disease risk analysis, resulting in the medical benefits of collected healthcare data being hardly to excavate [34]. For example, the shared ciphertext is mixed with a small portion of ECG signals, muscle signals, calorie data, and blood glucose, which is stored on the cloud server and can be accessed by healthcare service providers. All of the shared ciphertext needs to be decrypted at first, among which the small portion of ECG signals could be learned and analyzed for cardiovascular-related service provision [39]. Indeed, for cardiovascular professionals, resources are unnecessarily consumed for decryption and medical analysis efficiency is severely affected. Additionally, the encryption computation and ciphertext storage of CP-ABE are high resource demanding, which increases with the number of attributes in the access policy and brings severe resource consumption for resource-limited e-healthcare devices [1, 3, 22]. If amount of computation and storage resources are occupied for encryption, then e-healthcare devices may not have sufficient resources for accurate health data monitoring [35]. The resulting high energy consumption not only shortens the operation lifespan of e-healthcare devices but also releases enormous heat to impact patients' experience.

To address the above challenges, a novel privacy-preserving health data sharing framework is required for effective service provision and data utilization with cost-efficient resource consumption. Fog computing is a promising solution to assist data fusion, filtering, and analysis [24] in e-healthcare systems, since it extends data computing from a cloud to the edge of a network [25] and is more intelligent and powerful [5, 19] than e-healthcare devices. In our previous fog-assisted data sharing scheme [29], the fog node is integrated to process and re-encrypt the shared data for efficient medical analysis. Specifically, the scheme inherits both advantages from fog computing for efficient data pre-processing and CP-ABE for privacy preservation. However, the following issues still remain unaddressed. (1) How do we pre-process health data on the fog node for efficient data utilization? (2) How can patients retrieve and decrypt their health data after the fog node encrypts them? (3) How can we guarantee patient privacy when the shared data are re-encrypted by the semi-trusted fog node? (4) How do we prevent unauthorized data access if the fog node colludes with other entities for the shared plaintext?

In this article, we propose a privacy-preserving fog-assisted health data sharing (PFHDS) scheme to enhance data utilization efficiency and support effective medical service provision with privacy-preservation. First, patients encrypt their shared data with a personal access policy according to their interests and experiences, and the ciphertext is sent to the fog node. For efficient data utilization, the fog node classifies the collected health data into different categories of disease risks based on naive Bayes classification. With regard to every category of disease risk, the fog node defines specific attributes to encrypt the corresponding health items according to the healthcare background. Finally, the new ciphertext is transmitted to the cloud storage, and service providers can decrypt the ciphertext for effective healthcare service provision. To our best of knowledge, PFHDS is the first health data sharing scheme to integrate fog computing exquisitely for assisting the health data classification and data encryption in e-healthcare systems. Specifically, we list contributions of PFHDS as follows.

- We construct a secure fog-assisted health data sharing framework. Through a personalized access policy definition from patients, patients can retrieve their health data, and the shared data can be protected from disclosure during data processing on the fog node.

- We guarantee fine-grained access control with effective medical service provision. Through encryption with specific attributes on intelligent fog node with medical background, the shared data originally encrypted on patients can be accessed by authorized service providers with effective medical service provision.
- We achieve efficient data utilization for various kinds of healthcare service providers. The raw shared data are classified into different categories of disease risks, as well as corresponding health items regarding to disease risks are encrypted with attributes of professional healthcare service providers. As a result, the health data can be efficiently accessed and utilized, and hierarchical diagnosis and treatment can be achieved through efficient data sharing.
- Security analysis shows that the proposed scheme preserves the privacy of patients' health data and guarantees authorized data access during health data sharing, as well as resists collusions of fog node with other unauthorized entities. Furthermore, we conduct extensive simulations to demonstrate that PFHDS can achieve effective healthcare service provision and efficient data analysis with acceptable computational overhead.

The remainder of this article is organized as follows. Section 2 reviews the related works on data sharing for e-healthcare systems, and Section 3 introduces preliminaries and notations. Then, we present models and goals in Section 4. PFHDS details can be seen in Section 5. The security discussions and performance evaluations are presented in Sections 6 and 7, respectively, followed by a conclusion in Section 8.

2 RELATED WORKS

Existing health data sharing schemes are widely proposed that focus on two aspects: privacy preservation and efficiency.

Since health data are privacy sensitive, privacy preservation is critically researched in existing data sharing schemes. Chen et al. [6] proposed cloudlet-based health data sharing, which utilized Number Theory Research Unit (NTRU) to encrypt a user's body data from wearable devices and presented a trust model to help similar patients to communicate with each other about their diseases. Tong et al. [30] integrated attribute-based encryption with threshold signing for providing role-based access control with audibility to prevent potential misbehavior in e-healthcare systems. Yang et al. [37] proposed a medical record sharing scheme for cloud computing based on the classification of the attributes for medical records, which used vertical partitions of a medical dataset for different parts of medical data to achieve different privacy levels. Huang et al. [11] proposed a fine-grained electronic health records sharing scheme via similarity-based recommendation accelerated by Locality Sensitive Hashing in cloud-assisted e-healthcare systems. Yang et al. [38] proposed a data sharing scheme to a certain group of people in cloud-based multimedia systems to achieve privacy preservation in a particular time period. To solve the high computation challenge in secure data sharing, Li et al. [15] proposed to eliminate a majority of the computation task by adding system public parameters besides moving partial encryption computation offline. In addition, a public ciphertext test phase is performed before the decryption phase, which eliminates most of computation overhead due to illegitimate ciphertexts. For the sake of data security, a Chameleon hash function is used to generate an immediate ciphertext, which will be blinded by the offline ciphertexts to obtain the final online ciphertexts. To preserve location privacy during data sharing, Zhang et al. [44] proposed an enhanced privacy-preserving data sharing scheme through caching and spatial-anonymity (CSKA) in continuous LBSs, which adopted multi-level caching to reduce the risk of exposure of users' information to untrusted LSPs. Shen et al. [27] proposed a traceable group data sharing scheme by leveraging the key agreement and the group signature to support anonymous multiple users in public clouds. In this work, group members can communicate anonymously with respect to the group signature, and the real identities of members can be traced if necessary. However, a common conference key is derived based on the key agreement to enable group members to share and store their data securely. Yin et al. [40] proposed a privacy-enhanced data sharing scheme by allowing the data user to generate random query trapdoor. Through leveraging bloom filter and bilinear pairing operation to construct secure index for each data file, this proposed scheme can enable the cloud to perform data sharing without obtaining any useful information. Kang et al. [12] exploited consortium blockchain and smart contract technologies to achieve authorized data sharing in vehicular edge networks efficiently, as well as proposed a reputation based data sharing scheme to ensure high-quality data sharing among vehicles.

Efficiency in health data sharing is also widely concerned for the resource-limited e-healthcare devices. Chu et al. [7] proposed a new public-key crypto system that produced constant-size ciphertexts for secure data sharing, which can realize efficient delegation of decryption rights for any set of ciphertexts by aggregating secret keys into a single key. Wang et al. [31] proposed an efficient hierarchy attribute-based encryption scheme in cloud computing, which integrated the access tree with different security levels into one for various kinds of health files. Li et al. [16] divided the users in PHR systems into multiple security domains, which can reduce key management complexity for owners and users to achieve fine-grained and scalable data access control. Liu et al. [21] proposed an online/offline attribute-based encryption health data sharing scheme to reduce the encryption cost in mobile healthcare systems, which performed a majority of computation tasks on an offline phase, and an online phase can rapidly assemble the final ciphertext when electronic health records are known. Some approaches also proposed hybrid clouds to offload the encryption workload to a private cloud. Some propose private hybrid schemes. In Reference [8], Dan et al. propose to store sensitive data on a private cloud and less sensitive data on a public cloud to achieve storage elasticity as well as control over enterprise data. In Reference [14], Li et al. introduce a private cloud as an interface between users and a public cloud, as well as to manage private keys for users' privileges in the private cloud.

In summary, existing data sharing schemes neglected effective healthcare service provision and efficient data analysis, which are critical issues for valuable data utilization of health data in e-healthcare systems. Benefited from fog computing promisingly applied into e-healthcare systems, we propose a novel efficient and privacy-preserving fog-assisted data sharing scheme in e-healthcare systems.

3 PRELIMINARIES AND NOTATIONS

In this section, we briefly introduce some preliminaries on Shamir Secret Sharing, Bilinear maps and Naive Bayes classifier, as well as some important notations frequently used throughout the article in Table 1.

3.1 Shamir Secret Sharing

Shamir Secret Sharing scheme (t, n) [26] divides a secret s into n pieces $s_1, \ldots, s_n$, and satisfies the following two conditions:

(1) Any t or more than t pieces of s_i make secret s easily computable;
(2) Any $t - 1$ or fewer than t pieces of s_i leave s completely undetermined (in the sense that all its possible values are equally likely).

Table 1. Notations

H	Abnormal health dataset
(x, y)	The yth node in the xth level of access tree
index(x, y)	Unique value associated with node (x, y)
$parent_{(x,y)}$	Parent node of (x, y)
$q_{(x,y)}$	Polynomial for node (x, y)
D_i	The ith disease risk
SID_i	The ith health item indexes set related to D_i
ID_i	The ith health item name
P_j	The probability of disease risk D_j happens
P_{ji}	The probability of abnormal health item ID_i happens under the condition of disease D_j happens
A_{tree}	Personal access tree with root node A
B_{tree}	Professional access tree with root node B
DR_i	ith child of node B
$Dec(x, y)$	the decryption value of the node (x, y)
...	...

Shamir Secret Sharing scheme consists of the following two parts:

- ***Secret sharing.*** Suppose a user has a secret s to share with n parties. The user chooses a polynomial $f(x)$ with a degree k, i.e.,

$$f(x) = a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x + s, \tag{1}$$

where $a_1, \ldots, a_k$ are randomly chosen. The user generates $s_i = f(i)(i = 1, \ldots, n)$ and transmits s_i to n parties.

- ***Secret retrieval.*** More than t parties with secrets s_i retrieve the secret s. First, for all the index i in t parties, the Lagrange coefficient $\Delta_{i,j}(0)$ can be computed as follows:

$$\Delta_{i,j}(0) = \prod_{j=1, j \neq i}^{t} \frac{-j}{i-j}. \tag{2}$$

Second, a lagrange interpolation is used to calculate s from t pieces $s_1, \ldots, s_t$,

$$s = \sum_{i=1}^{t} f(i) \Delta_{i,j}(0). \tag{3}$$

3.2 Bilinear Maps

The bilinear pairings namely Weil pairing and Tate paring of algebraic curves are defined as a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$, where $\mathbb{G}_1$ is a cyclic additive group generated by g, whose order is a prime p, and $\mathbb{G}_T$ is a cyclic multiplicative group of the same order q. Discrete logarithm problems (DLP) in both $\mathbb{G}_1$ and $\mathbb{G}_T$ are hard. Bilinear pairings have the following properties:

- Bilinearity: for any $u, v \in \mathbb{G}_1$, and $a, b \in \mathbb{Z}_p$, it has $e(u^a, v^b) = e(u, v)^{ab}$;
- Non-degeneracy: $e(g, g) \neq 1$, 1 is the unit parameter in $\mathbb{G}_T$.
- Computability: for all $u, v \in \mathbb{G}_1$, there is an efficient algorithm to compute $e(u, v)$.

3.3 Naive Bayes Classification

Naive Bayes classification is used to classify the problem instance into one classifier based on the joint probability. Given k classifiers $(c_i, \ldots, c_k)$ and the problem instance $X = (x_1, \ldots, x_n)$ with

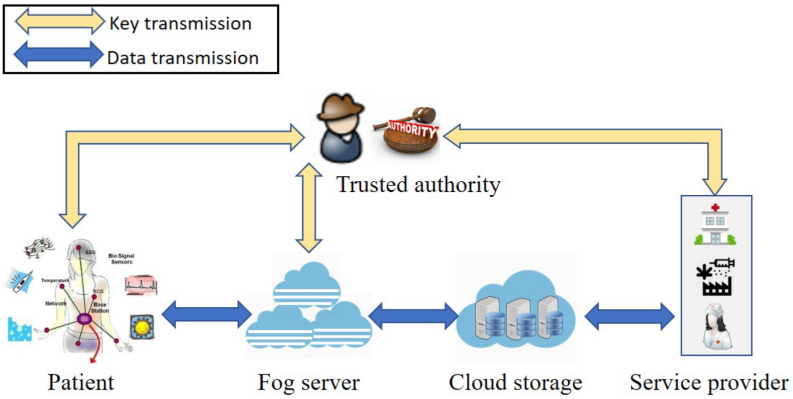


Fig. 1. Fog-assisted health data sharing system model.

n features. Compute the probability of c_i happens under the condition of X happens, which can be termed as $p(c_i|X)$. The corresponding classifier with maximum $p(c_i|X)$ can be taken as the predicted classifier.

According to Bayes's theorem, $P(c_i|X) = P(c_i, X)/P(X)$. For every $P(c_i|X)$, the values of feature X is given, such that $P(X)$ is effectively constant. As a result, $P(c_i|X) = P(c_i, X)/P(X)$ is equivalent to $P(c_i, X)$.

4 MODELS AND GOALS

In this section, we present the system model and the security model, as well as design goals.

4.1 System Model

The system consists of five entities: trusted authority, patients, fog node, cloud storage, and service providers as shown in Figure 1.

Trusted authority initializes the system, provides registration service, generates system public keys, system master keys, and secret keys for other entities.

Patients share the health data (e.g., heart rate and blood pressure), which can be collected by e-healthcare devices or manually input by themselves. Patients encrypt their shared data and send the ciphertext to a fog node.

fog node can be a health gateway or a router in physical proximity of patients. fog node masters healthcare background and powerful computation capabilities. It pre-processes and re-encrypts the shared ciphertext and then transmits the new ciphertext to a cloud storage.

Cloud storage is a remote third-party server that has powerful storage capabilities. It stores and manages the shared ciphertext transmitted from the fog node.

Service providers can be doctors, researchers, insurance companies, and drug manufacturers. Service providers use their attributes to access the shared ciphertext for learning health data and providing healthcare services.

4.2 Security Model

In the system, the trusted authority is fully trusted by all other entities, and the transmission channel among patients, fog node, and cloud storage is secure. The patient is trusted and aims to prevent unauthorized entities from obtaining the shared plaintext. The fog node and the cloud storage are *honest-but-curious*. The fog node provides data pre-processing services, and the cloud performs data storage, but both of them are curious about the shared plaintext. We categorize the security threats into unauthorized data access and collusion attack as follows. **(1) *Unauthorized data access.*** The unauthorized service providers may obtain the shared health data to spread advertisements and drug promotions. **(2) *Collusion attack.*** The fog node, cloud storage, and unauthorized service providers cannot learn the shared plaintext separately but they may collude with each other and intend to learn the shared plaintext.

4.3 Design Goals

In this article, PFHDS should achieve the following security goals and performance goals:

- **Security goals**
  (1) Data confidentiality. PFHDS aims to preserve data confidentiality for the patient. During the shared dataflows from the patient, through the fog node and cloud storage to the service provider, the shared data should be kept confidential from all the unauthorized entities.
  (2) Patient-centric access control. PFHDS aims to enable patient-centric access control with effective medical service provision. Meanwhile, PFHDS aims to enable the shared data to be retrieved and decrypted by the patient, although it is re-encrypted by the fog node and stored on the cloud storage.
  (3) Collusion attack resistance. PFHDS aims to resist collusion from unauthorized entities to obtain the shared plaintext, i.e., even the fog node, the cloud storage and service providers that without enough attributes cooperate with each other, they cannot obtain the shared data.
- **Performance goals**
  (1) Effective medical service provision. PFHDS aims to enable the shared data to be accessed by the service provider who has specific medical knowledge to analyze it, such that the patient may obtain effective medical services through data sharing.
  (2) Data classification. PFHDS aims to classify the shared data collected from heterogenous devices into categories in terms of disease risks, such that the shared data can be utilized by different service providers efficiently and the medical value can be excavated adequately.
  (3) Cost-efficient encryption. PFHDS aims to reduce the encryption burden in terms of computation, storage, and energy consumption on the patient with resource-limited e-healthcare devices.

5 FOG-ASSISTED HEALTH DATA SHARING SCHEME

5.1 Scheme Overview

To efficiently share the collected data in e-healthcare systems for effective medical service provision and efficient data analysis with privacy preservation, we design the PFHDS scheme. Figure 2 illustrates the overview of the proposed scheme. First, the shared data are collected from e-healthcare devices (e.g., wearable devices and bio-sensors) or the patient's manual inputs, and the patient encrypts the shared data and then transmits the ciphertext to the fog node. Second, the fog node pre-processes the health data and classifies the health data into categories after disease risk analysis with naive Bayes classification, as well as indexes health items for different disease risks. Third, the fog node re-encrypts the shared data with a new access policy according to disease risks and then transmits the ciphertext to the cloud storage. Finally, the service provider with authorized attributes accesses the ciphertext and decrypts it.

PFHDS consists of the following algorithms: *Setup*, *KeyGen*, *Encrypt*, *Pre-process*, *Re-encrypt*, and *Decrypt*.

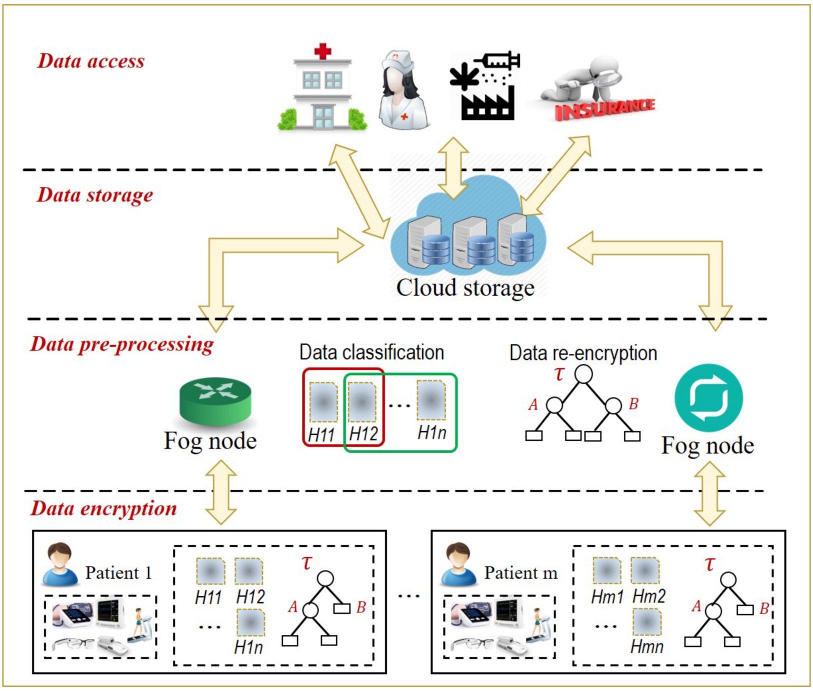


Fig. 2. Scheme overview of PFHDS.

- $Setup(U, \lambda) \rightarrow (PK, MSK)$. The trusted authority inputs universal attributes U and the security parameter λ, outputs the system public key PK and the system master key MSK.
- $KeyGen(PK, MSK, S, GLI_M, GLI_f) \rightarrow (SK, PK_f, SK_f)$. The trusted authority inputs the system public key PK, master key MSK, and attribute set S of the service provider, as well as the geographical location information GLI_M of the service provider and GLI_f of the fog node, and then outputs the secret key SK for the service provider, the public key PK_f, and the secret key SK_f for the fog node.
- $Encrypt(M) \rightarrow (CT)$. The patient encrypts the shared data M by using the symmetric encryption with a content key. Meanwhile, the patient defines a personal access policy according to his or her experiences and interests to encrypt the content key with attribute-based encryption and outputs the ciphertext CT.
- $Pre\text{-}process(CT) \rightarrow (D, SID)$. The fog node inputs the shared ciphertext, computes probabilities of disease risks according to the shared data, and outputs the *top-k* disease risks D as well as it corresponding health item set SID.
- $Re\text{-}encrypt(CT) \rightarrow (CT')$. The fog node defines a new access policy termed as a professional access policy τ according to *top-k* disease risks and re-encrypts the shared data with access policy τ. Meanwhile, the fog node encrypts indexes of related health items with different attributes for different disease risks. The fog node outputs the new ciphertext CT' and transmits it to the cloud storage.
- $DecryptNode(CT', S, PK, SK) \rightarrow (H)$. The service provider with attributes that satisfy the access policies defined by the fog node and the patient can use the public key PK and his or her secret key SK to decrypt the ciphertext CT' and learn the shared plaintext H.

5.2 Scheme Details

In this subsection, we construct the following five phases in PFHDS by using the above algorithms.

Phase* 1 *System initiation

The system initiation phase includes algorithms *Setup* and *KeyGen*. In this phase, the trusted authority sets up the system and generates keys for the fog node and the service provider.

The trusted authority takes as input the attributes U in the system. It chooses two multiplicative groups $\mathbb{G}$ and $\mathbb{G_T}$ of prime order p and a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G_T}$ between them. A generator g and u random group elements $h_1, h_2, \ldots, h_u \in \mathbb{G}$ that are associated with universal attributes U in the system. The set of attributes, i.e., $h_1, \ldots, h_u$ are common for all providers but are not kept secret by the trusted authority. They are associated with universal attributes in the system and are elements of public keys. In addition, the trusted authority chooses random exponents α, $a \in \mathbb{Z}_p$. The trusted authority outputs the public key PK and the system master key MSK,

$$PK = g, g^a, e(g, g)^\alpha, h_1, \ldots, h_u, \tag{4}$$

$$MSK = g^\alpha. \tag{5}$$

The trusted authority runs *KeyGen* algorithm to generate secret keys for the service provider. The service provider sends his or her attributes set S (including private attributes and professional attributes) to the trusted authority. The trusted authority takes the system master key MSK as input, selects a random $t \in \mathbb{Z}_p$, and then computes the secret key SK using the public keys $h_x(x \in S)$ that corresponding to the attributes for the service provider. Here, $SK = (K, K_x)$, where $x \in S$,

$$K = g^\alpha g^{at}, \forall x \in S\ K_x = h_x^t g^t. \tag{6}$$

The trusted authority generates the public key and the secret key for the fog node. After receiving the geographical location information GLI_f of the fog node, the trusted authority outputs public key PK_f as well as randomly chooses $V_f \in \mathbb{G}$ as the secret key SK_f for the fog node. After receiving the geographical location information GLI_M of the patient, the trusted authority sends the public key PK_f of his or her nearest fog node to the patient,

$$PK_f = U_f = e(V_f, g), \tag{7}$$

$$SK_f = V_f. \tag{8}$$

Phase* 2 *Data encryption with a personal access policy

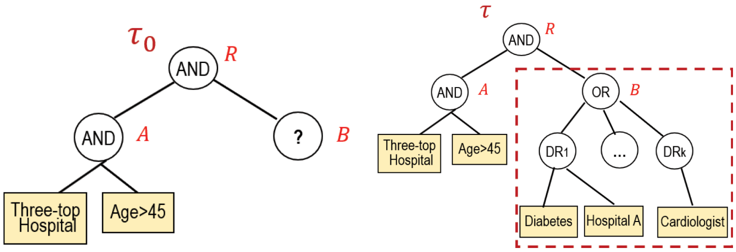
The patient runs *Encrypt* algorithm, which consists of two operations Enc_1 and Enc_2. First, the patient encrypts his or her shared data with a content key by using Enc_1. Second, the patient defines a personal access policy according to his or her experiences and interests, as well as integrates the attribute-based encryption to encrypt the content key of the shared data by using Enc_2.

The collected data are termed as $H' = \{H'_1, H'_2, \ldots, H'_n\}$. For valuable data sharing, the patient detects the abnormal health data. We consider that the patient stores a standard health item table $S_h = \{S_{h1}, S_{h2}, \ldots, S_{hn}\}$ as seen in Table 2(a). The collected data are compared with the standard value, and the abnormal health item is generated as $H = \{ID_1, ID_2, \ldots, ID_m\}$.

The patient encrypts the shared health data M by performing symmetric encryption (i.e., DES, AES) Enc_1 with a content key ck and computes the ciphertext $C_M = Enc_{1(ck)}(M)$. Here, the encrypted shared data M includes the profile and the specific value of the shared data.

For fine-grained data sharing, ck is encrypted as the following $Enc_2(PK, ck, \tau_0)$. The public key PK, content key ck, and an developing access tree τ_0 are taken as inputs. The patient selects a random number s in $\mathbb{Z}_p$ as the secret element for encryption. Then the patient computes $\hat{C}$ and $\tilde{C}$ as Equation (9),

$$\hat{C} = ck\ e(g, g)^{\alpha s}, \tilde{C} = g^s. \tag{9}$$



(a) Access tree on patient

(b) Access tree on fog node

Fig. 3. Access tree construction.

To integrate both the personal access policy and the professional access policy into one access tree, the patient constructs a developing access tree τ_0 as seen in Figure 3(a). In τ_0, the root node R masters the whole access policy and has two children nodes A and B, which are root nodes for personal access policy A_{tree} and professional access policy B_{tree}, respectively. The A_{tree} can be specifically constructed by the patient.

The access tree is constructed by 2 parts: tree nodes and polynomials. Here, we use (x, y) to represent every node, x represents the node is in the xth level, and y represents the node is yth node in xth level. To construct a tree structure rule, a polynomial $q_{(x,y)}$ is selected for every non-leaf node (x, y). In an access tree τ_0, the root node corresponds to the secret element s, other non-leaf nodes corresponds to a threshold, and leaf nodes corresponds to the required attributes. We hide the secret element s in the leaf nodes of the access tree. This process corresponds to the secret sharing process of the Shamir's secret sharing. We divide the value of every node to its children nodes according to the Shamir's secret sharing from top of the access tree to the bottom. In this way, we hide s in the children nodes.

Beginning from the root node R, the patient sets $q_R(0) = s$ and chooses one point of the polynomial q_R to define it completely. For each non-root node (x, y), it sets $q_{(x,y)}(0) = q_{parent_{(x,y)}}(index(x, y))$, where $index(x, y)$ returns an unique value associated with node (x, y) and $parent_{(x,y)}$ is the parent node of (x, y). After the tree traversal of τ_0, every node (x, y) has its own value of $q_{(x,y)}(0)$, which is the secret element for this node. Specifically, the secret s is distributed and hidden in all of the leaf nodes in the access tree.

For the following description, we term $s_1 = q_A(0)$ and $s_2 = q_B(0)$, i.e., s_1 is the secret element for A_{tree}, and s_2 is the secret element for B_{tree}).

Let Y_A be the set of leaf nodes in A_{tree}. The patient computes $C_{A(x,y)}$ and $C'_{A(x,y)}$ for all nodes (x, y) in set Y_A as Equation (10),

$$C_{A(x,y)} = g^{aq_{(x,y)}(0)}, C'_{A(x,y)} = h_{(x,y)}^{q_{(x,y)}(0)}. \tag{10}$$

To securely construct B_{tree} on the fog node, the patient selects a random $r_i \in Z_p$ and encrypts s_2 with the system public key as well as the fog node's public key as Equation (11),

$$C_{s_2} = \left(g^{r_i}, s_2 U_f^{r_i}\right). \tag{11}$$

Finally, the patient sends the encrypted shared data $(C_M, \hat{C}, \tilde{C}, C_{A(x,y)}, C'_{A(x,y)}, \tau_0, C_{s_2})$ to the fog node.

Phase 3 Data pre-processing

The fog node runs *Pre-process* algorithm, which takes the ciphertext transmitted from the patient as input, and outputs the pre-processing result disease risks S, as well as its corresponding health item indexes set SID.

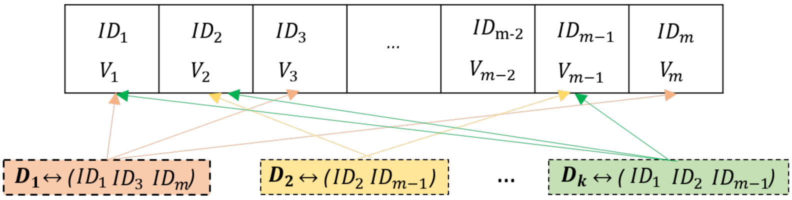

ID_1
V_1
ID_2
V_2
ID_3
V_3
...
ID_{m-2}
V_{m-2}
ID_{m-1}
V_{m-1}
ID_m
V_m
$D_1 \leftrightarrow (ID_1\ ID_3\ ID_m)$
$D_2 \leftrightarrow (ID_2\ ID_{m-1})$
...
$D_k \leftrightarrow (ID_1\ ID_2\ ID_{m-1})$

Fig. 4. *Top-k* disease risks and related health item indexes. This example shows there are k disease risks analyzed: D_1 is related to (ID_1, ID_3, ID_m) health items; D_2 is related to (ID_2, ID_{m-1}) health items; D_k is related to (ID_1, ID_2, ID_{m-1}) health items.

Table 2. Health Data Sheet

(a) Standard Health Value

Health item	Value
Systolic pressure	90-140
Diastolic pressure	60-90
Heart rate	60-100
...	...

(b) Probability of Disease

	ID_1	...	ID_m
D_1	P_{11}	...	P_{1m}
D_2	P_{21}	...	P_{2m}
...	...	...	...
D_q	P_{q1}	...	P_{qm}

We consider that the well-trained classification model computes relations between disease risks and health items according to Reference [20] in Table 2(b), where $P_{ji} = P(ID_i|D_j)$ represents the probability that abnormal health item ID_i happens under the condition of disease D_j. The fog node computes probabilities P_j of all disease risks based on naive Bayes classification [4] as seen in Equation (12) using Table 2(b) and then sorts the probabilities to obtain *top-k* health disease risks, as well as indexes the related health items for every disease risk.

The pre-processing of health data on fog node can be detailed as Algorithm 1. The health item indexes set SID_i corresponding to disease risk D_i can be seen in Figure 4:

$$\begin{aligned} P_j &= P(D_j|H) \\ &= P(D_j, H)/P(H) \\ &= P(D_j, H)^{*} \\ &= P(D_j, ID_1, \ldots, ID_m) \\ &= P(D_j)\prod_{i=1}^{m} P(ID_i|D_j) \\ &= P(D_j)\prod_{i=1}^{m} P_{ji}. \end{aligned} \tag{12}$$

Phase 4 Re-encryption with a professional access policy

To enable effective medical service provision for the patient according to his or her disease risks, the fog node runs *Re-encrypt* algorithm to re-encrypt the health data with a professional access policy. Meanwhile, for efficient data analysis, the fog node encrypts the related health item indexes for every disease risk.

After the *top-k* possible disease risks are analyzed, the fog node searches from its own storage; if there are corresponding attributes that related to the possible disease risk, then the fog

[*] For every P_j, the values of feature H is given, such that $P(H)$ is constant and can be eliminated.

ALGORITHM 1: Pre-processing Algorithm

```
Input: $C_H$ and $T_d$
Output: Set $D$ and set $SID$, where $D = \{D_1, D_2, \ldots, D_k\}$ and $SID = \{SID_1, SID_2, \ldots, SID_k\}$,
        and the corresponding $SID_i$ is the health item indexes set related to $D_i$ ($i$ is from 1 to $k$)
D = null;
SID = null;
for $i = 1$ to $q$ do
    Compute the probability $P_i$ of disease risk $D_i$ happens according to Equation (12);
    Add $(P_i, D_i)$ into a set $P$;
end
Sort set $P$ according to the probabilities $P_i$ from the largest to the smallest;
for $j = 1$ to $k$ do
    Add the corresponding disease risk $D_j$ of the top-$k$ probability $P_j$ into the set $D$;
    for $t = 1$ to $m$ do
        if $ID_t$ is related to disease risk $D_j$ then
            Add health item $ID_t$ into the set $SID_j$;
        end
    end
end
Output set $D$ and set $SID$;
```

node sets professional attributes as professional access policy. Otherwise, the fog node requires cloud storage by providing the *top-k* possible diseases D_j ($j = 1$ to $j = k$) name, and the cloud storage returns the attributes that can provide effective health service for the specific disease risk, and the fog node stores professional attributes. For example, if the computed *top*-2 possible disease risks are $\{diabetes, heart\ disease\}$ and professional diabetes-related service can be provided in *HospitalA* or *HospitalC*, then the fog node sets the professional attributes as $\{Diabetes\ Doctor, Hospital\ A\ or\ Hospital\ C\}$ as the access policy for *diabetes* disease risk.

The fog node constructs B_{tree} to complement the access tree τ from the developing access tree τ_0. In B_{tree}, the root node B has k children nodes, where each node represents one possible disease risk analyzed through pre-processing and masters the attributes that can provide corresponding healthcare services. The leaf nodes of B_{tree} are professional attributes. The threshold of node B is 1, which means the service provider with attributes that can deal with only one of the k disease risks is able to satisfy B_{tree}. The developed access tree τ can be seen as Figure 3(b).

The secret element of node B is s_2, which should be sealed in B_{tree}. The fog node decrypts ciphertext $C_{s_2} = (g^{r_i}, s_2 U_f^{r_i})$ to obtain the secret element s_2. The decryption computation is as follows:

$$s_2 = \frac{s_2 U_f^{r_i}}{e(V_f, g^{r_i})}, \tag{13}$$

$$q_B(0) = s2. \tag{14}$$

Beginning from the root node B of B_{tree}, the fog node sets $q_B(0) = s_2$ and chooses one point of the polynomial q_B to define it completely. For all the nodes (x, y) except B in B_{tree}, the fog node sets $q(x, y)(0) = q_{parent_{(x,y)}}(index(x, y))$. After the tree traversal for B_{tree} from root to bottom, every node (x, y) has its own value of $q_{(x,y)}(0)$, which is the secret element for this node. The root node B has k children nodes, and every node denotes one kind of disease risk. Let the ith child of node B be the ith disease risk, termed node DR_i. Specifically, nodes $DR_1, \ldots, DR_k$ have their values of $q_{DR_i}(0)$ (where i is from 1 to k) as Equation (15),

$$q_{DR_i}(0) = q_B(index\ of\ DR_i). \tag{15}$$

To efficiently share related health items for different categories of service providers, the fog node encrypts the indexes of the health items. The fog node computes C_i and $\hat{C}_i$ for every SID_i as Equation (16), where i is from 1 to k,

$$C_i = SID_i e(g,g)^{\alpha q_{DR_i}(0)}, \tilde{C}_i = g^{q_{DR_i}(0)}. \tag{16}$$

Let Y_B be the set of leaf nodes in B_{tree}. The fog node computes $C_{B(x,y)}$ and $C'_{B(x,y)}$ for all nodes (x,y) in set Y_B as Equation (17),

$$C_{B(x,y)} = g^{aq_{(x,y)}(0)}, C'_{B(x,y)} = h_{(x,y)}^{q_{(x,y)}(0)}. \tag{17}$$

Then, the secret element s_2 of node B is sealed in $C_B(x,y)$ and $C'_{B(x,y)}$ in B_{tree}. For simplicity, since $C_A(x,y)$ and $C_B(x,y)$ have the same structure, we term them $C_{(x,y)}$. Meanwhile, since $C'_{A(x,y)}$ and $C'_{B(x,y)}$ have the same structure, we term them $C'_{(x,y)}$. As we can see, the secret element s for encryption of the content keys is sealed in $(C_{(x,y)}, C'_{(x,y)}, \tau)$.

After re-encryption, the fog node computes a new ciphertext $CT' = (C_M, \hat{C}, \tilde{C}, C_{(x,y)}, C'_{(x,y)}, \hat{C}_i, \tilde{C}_i, \tau)$ (where i is from 1 to k) and sends CT' to the cloud storage.

Phase* 5 *Data access

In this phase, the service provider runs $DecreyptNode$ algorithm and accesses the ciphertext. The service provider can decrypt the shared ciphertext only when his or her attributes satisfy the personal and professional access policies. Meanwhile, the service provider with attributes that satisfy the access policy of disease risk D_i can decrypt $\hat{C}_i$ and obtain the corresponding related health item index set SID_i. When a service provider possesses an attribute corresponds to a leaf node of the access tree, the service provider can decrypt the leaf node with his or her secret key. When a non-leaf node of the access tree with sufficient (≥ threshold nodes) nodes that can be decrypted, this node can be decrypted according to the retrieval process of Shamir's secret sharing.

A service provider with attribute set S needs the public key PK and his or her secret key SK to decrypt CT'. Let $DecryptNode(CT', S, SK, PK)$ be the operation, and $D_{(x,y)}$ be the decryption result for every node (x,y). To decrypt the ciphertext sealed in the access tree τ, a service provider runs $DecryptNode(CT', S, SK, PK)$ from bottom to top manner. There are two cases to decrypt the nodes in the access tree τ:

Case 1 : (x,y) is a leaf node:

If the attribute $att(x,y)$ represented by node (x,y) is $\notin S$, then $DecryptNode(CT, S, SK, PK, (x,y)) = null$. Otherwise, the operation $DecryptNode(CT, SK, PK, (x,y))$ is performed as Equation (18),

$$\begin{aligned} &DecryptNode(CT, SK, PK, (x,y)) \\ &= \frac{e(C_{(x,y)}, K_x)}{e(C'_{(x,y)}, g^a)} \\ &= \frac{e(g^a q_{(x,y)}(0), h_x g^t)}{e\left(h_{(x,y)}^{q_{(x,y)}(0)}, g^a\right)} \\ &= e(g,g)^{atq_{(x,y)}(0)}. \end{aligned} \tag{18}$$

Case $2 : (x, y)$ is a non-leaf node:

A service provider runs $DecryptNode(CT', S, SK, PK, (x, y))$ recursively. For all nodes z that are children of (x, y), it runs $DecryptNode(CT', S, SK, PK, z)$ and stores the output as Dec_z. Let $Z_{(x,y)}$ be an arbitrary $s_{(x,y)} - sized$ children nodes set of node (x, y), and $s_{(x,y)}$ is the threshold of node (x, y). Since the decryption is run from bottom to top manner, Dec_z of children nodes z in set $Z_{(x,y)}$ is computed before computing $Dec_{(x,y)}$.

For all nodes in $Z_{(x,y)}$, let $i = index(z)$ and $S_z = \{index(z) : z \in Z_{(x,y)}\}$. According to the polynomial structure rule, Dec_z can be written as Equation (19). $q_{(x,y)}(0)$ can be computed by using the Lagrange interpolation method as Equation (20), and $Dec_{(x,y)}$ of the node (x, y) can be computed as Equation (21):

$$\begin{aligned} Dec_z &= e(g,g)^{atq_z(0)} \\ &= e(g,g)^{atq_{(x,y)}(i)}, \end{aligned} \tag{19}$$

$$q_{(x,y)}(0) = \sum q_{(x,y)}(i)\Delta_{i,S_z}(0), \tag{20}$$

$$\begin{aligned} Dec_{(x,y)} &= \prod_{z \in Z_{(x,y)}} Dec_z^{\Delta_{i,S_z}(0)} \\ &= \prod_{z \in Z_{(x,y)}} e(g,g)^{atq_{(x,y)}(i)\Delta_{i,S_z}(0)} \\ &= e(g,g)^{at \sum q_{(x,y)}(i)\Delta_{i,S_z}(0)} \\ &= e(g,g)^{atq_{(x,y)}(0)}. \end{aligned} \tag{21}$$

After above operations, when a service provider with attributes that satisfy the personal access policy A_{tree}, the service provider can compute Dec_A as in Equation (22),

$$\begin{aligned} Dec_A &= e(g,g)^{atq_A(0)} \\ &= e(g,g)^{ats_1}. \end{aligned} \tag{22}$$

Specifically, when there is a service provider with attributes that satisfy the access policy of disease risk D_i, the service provider can compute Dec_{DR_i} as shown in Equation (23). The related health item indexes set SID_i can be decrypted as in Equation (24),

$$Dec_{DR_i} = e(g,g)^{atq_{DR_i}(0)}, \tag{23}$$

$$\begin{aligned} SID_i &= \frac{\hat{C}_i Dec_{DR_i}}{e(k, \tilde{C}_i)} \\ &= \frac{SID_i e(g,g)^{q_{DR_i}(0)} e(g,g)^{atDR_i(0)}}{e(g^\alpha g^{at}, g^{DR_i}(0))}. \end{aligned} \tag{24}$$

When there is a service provider with attributes that satisfy the professional access policy B_{tree}, the service provider can compute Dec_B as Equation (25),

$$\begin{aligned} Dec_B &= e(g,g)^{atq_B(0)} \\ &= e(g,g)^{ats_2}. \end{aligned} \tag{25}$$

Here, we represent the index of node A as x_1 and the index of node B as x_2, which means $s_1 = q_R(x_1)$ and $s_2 = q_R(x_2)$. According to the Lagrange interpolation, we can compute the encryption secret element s as Equation (26),

$$\begin{aligned} s &= q_R(x_1)\frac{-x_2}{x_1 - x_2} + q_R(x_2)\frac{-x_1}{x_2 - x_1} \\ &= s_1\frac{-x_2}{x_1 - x_2} + s_2\frac{-x_1}{x_2 - x_1}. \end{aligned} \tag{26}$$

As seen in the access tree τ, the decryption for root node R requires the decryption for node A and node B. When there is a service provider with attributes that satisfy both personal access policy A_{tree} and professional access policy B_{tree}, the service provider can compute Dec_R as Equation (27). The content key ck can be decrypted with secret key k as Equation (28),

$$\begin{aligned} Dec_R &= Dec_A^{\Delta_{x_1,x_2}(0)} \times Dec_B^{\Delta_{x_2,x_1}(0)} \\ &= Dec_A^{\frac{-x_2}{x_1-x_2}} \times Dec_B^{\frac{-x_1}{x_2-x_1}} \\ &= e(g,g)^{at(s_1\frac{-x_2}{x_1-x_2}+s_2\frac{-x_1}{x_2-x_1})} \\ &= e(g,g)^{ats}, \end{aligned} \tag{27}$$

$$\begin{aligned} ck &= \frac{\hat{C}Dec_R}{e(k,\tilde{C})} \\ &= \frac{cke(g,g)^{\alpha s}e(g,g)^{ats}}{e(g^{\alpha}g^{at},g^{s})}. \end{aligned} \tag{28}$$

Finally, the shared ciphertext C_M can be decrypted with symmetric key ck to reveal shared health plaintext M.

Specifically, the patient who selects the secret element s randomly can retrieve his or her own shared data by using s and public key $e(g,g)^{\alpha}$ directly as Equation (29),

$$ck = \frac{\hat{C}}{e(g,g)^{\alpha s}}. \tag{29}$$

From the above decryption process, we can see that if there is a service provider with attributes that can satisfy both access policies defined by the patient and the fog node, then he or she can obtain the shared plaintext. Specifically, if there is a service provider with attributes that satisfy the access policy of disease risk D_i, then he or she can obtain related health items to provide healthcare analysis and services efficiently.

6 SECURITY DISCUSSIONS

In this section, we discuss the security properties of PFHDS. Specifically, we demonstrate that PFHDS can achieve *Data confidentiality*, *Patient-centric access control*, and *Collusion attack resistance.*

- *Data confidentiality*

The shared data are encrypted with a content key by using symmetric encryption. Meanwhile, the patient encrypts the content key with a personal access policy for secure data sharing. The fog node, cloud storage, and unauthorized service providers cannot decrypt the shared ciphertext without decryption key and sufficient attributes, such that PFHDS can keep shared data confidential from the *honest-but-curious* fog node and the cloud storage. Furthermore, since the encrypted data is transmitted through the secure channel from patients to the cloud storage through the fog node, PFHDS resists data tampering from other unauthorized entities.

• *Patient-centric access control*

We discuss patient-centric access control during health data sharing from the following two aspects.

(1) The patient can decide his or her shared data to be accessed by the health serivce provider according to his or her personal customization. The content key for the shared data is encrypted with the access tree τ, which includes access tree A_{tree}, which masters the personal access policy constructed by the patient according to his or her experiences and interests. Only the service provider with attributes that satisfy A_{tree} has the probability to satisfy the access tree τ and obtain the content key, such that the patient can decide his or her shared data to be accessed by the service provider that satisfies his or her own specific requirements.

(2) The patient can retrieve and decrypt the shared data after data re-encryption by the fog node. In PFHDS, since the secret element s for content key ck is selected randomly by patient himself, the patient can directly use s and the system public key $e(g,g)^{\alpha}$ to decrypt $\hat{C}$ and obtain the content key according to Equation (24). As a result, although the shared data is re-encrypted by the fog node, the patient can also retrieve his or her shared health data flexibly.

• *Collusion attack resistance*

PFHDS can resist the following collusions for obtaining the shared plaintext.

(1) Collusion between the fog node and the cloud storage. The shared data are kept confidential from the fog node and the cloud storage separately. The fog node has the ability to decrypt the secret element s_2 of B_{tree}, but it cannot learn the secret element s without attributes that satisfy the personal access policy that is constructed in A_{tree}. Cloud storage learns nothing from the shared ciphertext CT'. Even if the fog node and the cloud storage collude with each other, they cannot learn the shared plaintext.

(2) Collusion between the fog node and the service provider. We consider two cases as follows.

Case 1: The service provider has no attributes to satisfy the personal access policy in A_{tree}, i.e., he or she cannot compute $Dec_A = e(g,g)^{ats_1}$ according to Equation (22). As we can see in Figure 3(b), the access tree τ is the combination of A_{tree} and B_{tree}. Only if the service provider can compute both $Dec_A = e(g,g)^{ats_1}$ and $Dec_B = e(g,g)^{ats_2}$ can it decrypt Dec_R according to Equation (27), which is a significant step to compute the content key. According to Equation (13), the fog node has the ability to decrypt the secret element s_2 of B_{tree}. In this case, the service provider cannot compute $Dec_A = e(g,g)^{ats_1}$; we can absolutely demonstrate that the collusion of the fog node and service provider cannot learn the shared plaintext in this case.

Case 2: The service provider has attributes that satisfy the personal access policy in A_{tree}, i.e., he or she can compute $Dec_A = e(g,g)^{ats_1}$ according to Equation (22). The fog node can decrypt the secret element s_2 of B_{tree} according to Equation (13) with its private key. However, the random chosen exponent a in the system and the random chosen number t for the specific service provider are only known by the authority and are not known by the service provider and the fog node. The service provider cannot compute the secret element s_1 of A_{tree} from $Dec_A = e(g,g)^{ats_1}$, which can be used with s_2 to compute s according to Equation (26) and then compute the content key ck according to Equation (29). Meanwhile, the fog node cannot compute $Dec_B = e(g,g)^{ats_2}$ from s_2, which can be used with $Dec_A = e(g,g)^{ats_1}$ to compute $Dec_R = e(g,g)^{ats}$ according to Equation (27) and then compute the content key ck according to Equation (28). As a result, collusion of the fog node and the service provider cannot obtain the shared data plaintext.

(3) Collusion among service providers. For this kind of collusion, we term the colluding service providers as U_1 and U_2 (we consider two colluding service providers for simplicity). The attribute set of service provider U_1 is A_1, and the attribute set of service provider U_2 is A_2. The random secret element of U_1 is t_1 in *KeyGen* algorithm run by the trusted authority, and the random secret

element of U_2 is t_2 in *KeyGen* algorithm run by the trusted authority. We can analyze this collusion from two the following cases.

Case 1: The union set $A_1 \cup A_2$ does not satisfy the access policies. It is obvious that this case of collusion cannot decrypt the ciphertext to learn the shared data plaintext.

Case 2: The union set $A_1 \cup A_2$ satisfies the access policies, but neither A_1 nor A_2 satisfies the access policies. For the node (x, y) that needs both attributes from A_1 and attributes from A_2 to compute $Dec_{(x,y)}$, let set I be the children nodes of (x, y) corresponding to the attributes from A_1; similarly, let set J be the children nodes of (x, y) corresponding to the attributes from A_2. For simplicity of illustration, we consider there is only one node in set I and set J, and the node is I and J, respectively. The decryption of node I is $Dec_I = e(g, g)^{at_1 q_I(0)}$ and the decryption of node J is $Dec_J = e(g, g)^{at_2 q_J(0)}$. The index of node I is i and the index of node J is j.

According to Equation (27), the decryption $Dec_{(x,y)}$ of node (x, y) is computed as follows:

$$\begin{aligned} Dec_{(x,y)} &= Dec_I^{\Delta_{i,j}(0)} \times Dec_J^{\Delta_{j,i}(0)} \\ &= e(g,g)^{at_1 q_I(0)\Delta_{i,j}(0)} \times e(g,g)^{at_2 q_J(0)\Delta_{j,i}(0)} \\ &= e(g,g)^{a(t_1 q_{(x,y)}(i)\Delta_{i,j}(0)+t_2 q_{(x,y)}(j)\Delta_{j,i}(0))}. \end{aligned} \tag{30}$$

According to Lagrange interpolation, we can compute $q_{(x,y)}(0)$ as follows:

$$q_{(x,y)}(0) = q_{(x,y)}(i)\Delta_{i,j}(0) + q_{(x,y)}(j)\Delta_{j,i}(0). \tag{31}$$

Generally, since t_1 and t_2 are randomly generated by the trusted authority, and are not known by the colluding service providers, we can analyze that $Dec_{(x,y)} \neq e(g, g)^{at_1 q_{(x,y)}(0)}$ and $Dec_{(x,y)} \neq e(g, g)^{at_2 q_{(x,y)}(0)}$ when $t_1 \neq t_2$. As a result, the colluding service providers cannot compute ck according to Equation (28) with $k_1 = g^\alpha g^{at_1}$ or $k_2 = g^\alpha g^{at_2}$ in this case.

7 PERFORMANCE EVALUATION

In this section, we evaluate the performance of PFHDS in terms of computation cost, storage cost, and the energy cost, which are computed on a 256-bit Bareto-Naehrig curve using version 0.3.1 of the RELIC library [2]. Since there are various kinds of e-healthcare devices equipped on the patient, we evaluate PFHDS on two typical e-healthcare platforms: a mobile phone and a sensor, respectively. The mobile phone has ARM Cortex-A9 CPU and 1 GB RAM, and the sensor has 32-MHz ARM Cortex-A3, 256 KB flash, and 32 KB RAM. The fog node has Intel Core i5 CPU and a RAM size of 4 GB. Times are measured in milliseconds (averaged over 10,000 iterations).

Let E_i (respectively, M_i) denote an exponentiation (respectively, multiplication) in the group $\mathbb{G}_i$. The bilinear operations are the dominate cost, such that we ignore minor factors such as arithmetic in $\mathbb{Z}_p$. $L_{\mathbb{G}_i}$ denotes the bit-length of the element in the group $\mathbb{G}_i$. U denotes the number of universal attributes. e denotes the paring time. P represents the number of attributes in the access policy that can decrypt the shared data; P_1 represents the number of attributes in the personal access policy; and P_2 represents the number of attributes in the professional access policy, where $P = P_1 + P_2$. Let $R = \frac{P_1}{P}$ be the percentage of personal access attributes in the whole access attributes. k denotes the number of health data categories on the fog node. The attributes in performance evaluation are selected from electronic health records and database field in Reference [9], which includes professional and personal information of healthcare service providers as well as patients, such as hospital, department, working years, age, city, and gender. These attributes are representative for e-healthcare systems. Since every attribute i is mapped to a public key h_i in group $\mathbb{G}$ before used in the expensive computations, the evaluation performance of encryption is not related to the specific attribute. The number of attributes involved in each instance of data encryption does not exceed 30 in most cases [32], such that we evaluate the performance with up to 30 attributes. We analyze

Table 3. Comparisons between CP-ABE and PFHDS

	CP-ABE	PFHDS
Encryption time on patient	$(2P+1)E_1$ $+E_T+M_T$	$(2P_1+3)E_1$ $+E_T+2M_T$
Re-Encryption time on fog node	–	$(2P_2+k)E_1+kE_T$ $+(k+1)M_T+e$
Whole encryption time	$(2P+1)E_1$ $+E_T+M_T$	$(2P+3+k)E_1$ $+(k+1)E_T$ $+(k+3)M_T+e$
Storage on patient	L_{G_T} $+(2P+1)L_{G_1}$	$2L_{G_T}$ $+(2P_1+2)L_{G_1}$
Storage on fog node	–	$(k+1)L_{G_T}$ $+(2P+k+2)L_{G_1}$

the theoretical results of PFHDS and CP-ABE [33] in terms of the encryption time and the storage cost as presented in Table 3, as well as demonstrate the performance from the computation cost, storage cost, and energy consumption as follows.

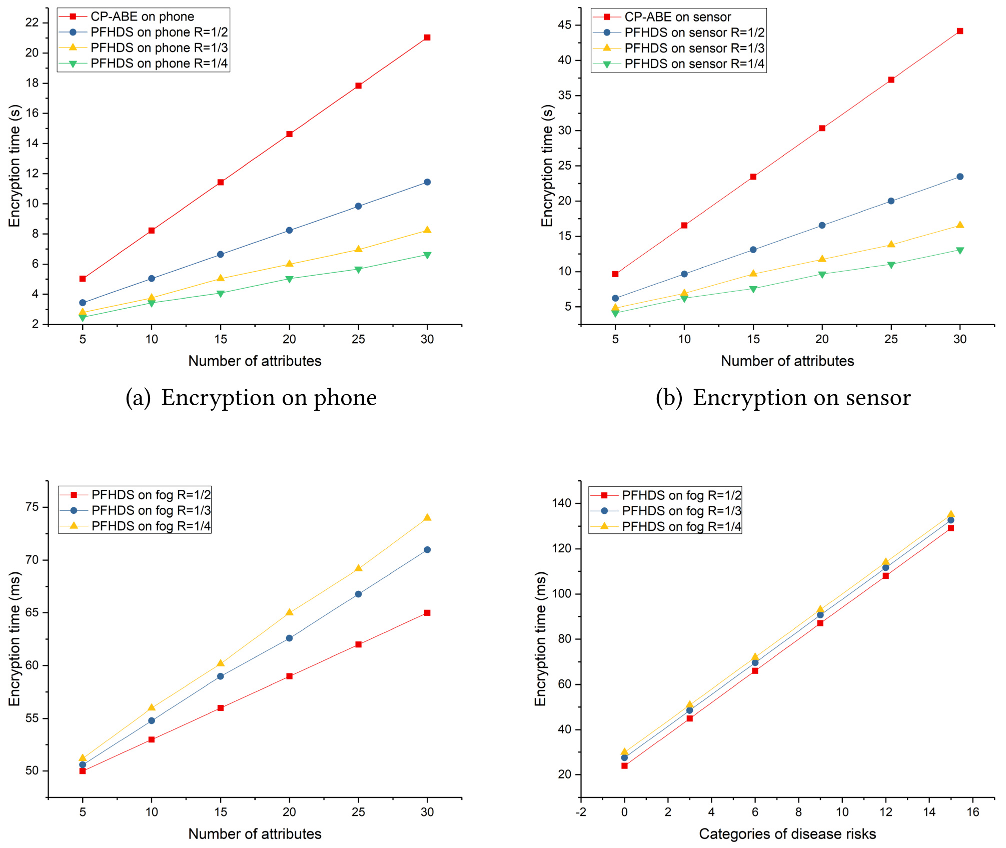
7.1 Computation Cost

We compare the computation cost on the mobile phone and the sensor between CP-ABE [33] and PFHDS, respectively, when the attribute percentage $R = 1/2, R = 1/3$, and $R = 1/4$ in Figure 5(a) and Figure 5(b). For the fog node, since it is not integrated in CP-ABE, the encryption time on the fog node of CP-ABE is 0; while the encryption time of PFHDS increases with the number of attributes P_2 for professional access policy B_{tree} and the number of categories k. In Figure 5(c), we set $k = 5$ (the shared data are classified into five categories on the fog node) and compare the encryption time on the fog node when $R = 1/2, R = 1/3$, and $R = 1/4$. In Figure 5(d), we set $P = 20$ (there are 20 attributes in the the whole access policy τ) and compare the encryption time on the fog node when $R = 1/2, R = 1/3$, and $R = 1/4$.

As shown in both of Figure 5(a) and Figure 5(b), the encryption time on the mobile phone and the sensor increases with the number of attrbutes. When the attribute percentage R decreases from 1/2 to 1/4, the encryption time consumed to encrypt the health data in PFHDS decreases successively. Meanwhile, the encryption time of PFHDS approximately equals R times of the encryption time of CP-ABE at the same number of attributes. Specifically, from Figure 5(a) and Figure 5(b), we can demonstrate that the encryption on sensor is more time-consuming than that on the mobile phone. The data encryption on sensor with CP-ABE approaches 30s when there are 20 attributes defined in the access policy, while PFHDS can only consumes 9s when $R = 1/4$ at the same attribute number, which improves significantly to reduce the time latency. As shown in Figure 5(c), the encryption time on the fog node increases with the number of attributes and increases when $R = 1/2$ decreases to $R = 1/4$ since there are appropriately $(1 - R)$ times of whole encryption is offloaded from the patient to the fog node. As demonstrated in Figure 5(d), the encryption time on the fog node also increases with categories of disease risks. When more disease risk categories are classified by the fog node, the fog node needs more computational resources, which is reasonable and acceptable since more efficient data utilization can be obtained by healthcare service providers.

7.2 Storage Cost

We evaluate the storage cost on the patient and the fog node, respectively. In Figure 6(a), we illustrate the relation between the number of attributes and the storage cost of CP-ABE and PFHDS



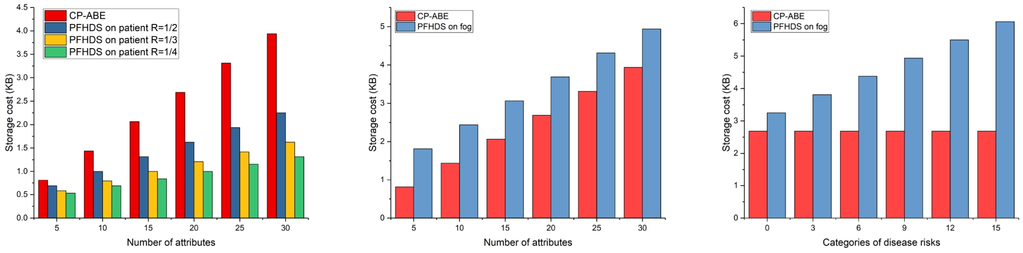
(a) Encryption on phone

(b) Encryption on sensor

(c) Encryption on fog (P)

(d) Encryption on fog (k)

Fig. 5. Encryption between CP-ABE and PFHDS.



(a) Ciphertext storage on patient

(b) Ciphertext storage on fog (P)

(c) Ciphertext storage on fog (k)

Fig. 6. Ciphertext storage between CP-ABE and PFHDS.

on the patient when $R = 1/2$, $R = 1/3$, and $R = 1/4$. For storage cost on the fog node, no ciphertext storage is needed in CP-ABE due to the disengagement of a fog node. In Figure 6(b), we set $k = 5$ and illustrate the relation between the number of attributes and the storage cost in CP-ABE and PFHDS. In Figure 6(c), we set $P = 20$ and illustrate the relation between categories of disease risks and the storage cost.

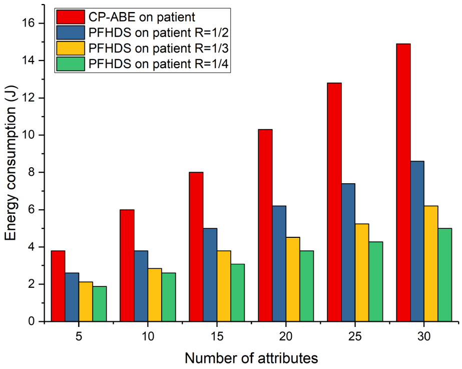


Fig. 7. Energy consumption between CP-ABE and PFHDS.

As shown in Figure 6(a), the storage cost on the patient increases with the number of attributes. The encryption time of PFHDS is less than and approximately equals R times of the encryption time in CP-ABE at the same number of attributes. Meanwhile, when the attribute percentage R decreases from 1/2 to 1/4, the storage cost to encrypt the health data consumed by PFHDS decreases successively. As demonstrated from Figure 6(b), the storage cost on the fog node and the whole storage cost on CP-ABE increases with the number of attributes. As seen in Figure 6(c), the fog node needs more storage cost when more disease risk categories are classified in PFHDS, while the whole storage cost in CP-ABE is stable when categories of disease risks increase. From both Figure 6(b) and Figure 6(c), the storage cost on the fog node in PHFDS is more than the whole storage cost on CP-ABE since the classified shared ciphertext requires extra storage resources, which is reasonable and acceptable while efficient data utilization can be obtained by healthcare service providers with health data classification.

7.3 Energy Consumption

Energy consumption is a major concern for encryption running on the patient that is equipped with resource-limited e-healthcare devices. For evaluating energy consumption, we employ PowerTutor to monitor energy consumption in PFHDS by using built-in battery voltage sensors and knowledge of battery discharge behavior [43]. First, we run various encryption applications with different attribute numbers in our evaluating mobile phone with ARM Cortex-A9 CPU and 1 GB RAM. Second, we collect energy consumption data shown on PowerTutor. In Figure 7, we illustrate the relation between the number of attributes (x-coordinate) and the energy consumption (y-coordinate) of CP-ABE and PFHDS on the mobile phone when $R = 1/2, R = 1/3$, and $R = 1/4$. We can demonstrate that the energy consumpiton of PFHDS is less than and approximately equals R times of the energy consumption of CP-ABE at the same number of attributes. Meanwhile, when the attribute percentage R decreases from 1/2 to 1/4, the energy consumption consumed by PFHDS decreases successively, since more encryption is offloaded from the patient to the fog node.

8 CONCLUSION

In this article, we have proposed PFHDS, which can achieve effective medical service provision and efficient data utilization with cost-efficient resource consumption. First, PFHDS supports efficient health service provision for patients due to the professional access policy by the fog node. Second, PFHDS enhances data analysis efficiency for service providers by classifying the health data into categories and indexing related health items. Third, PFHDS preserves health data privacy and access control for patients during health data sharing even under the collusion of the fog node and other entities. Finally, PFHDS reduces the resource consumption on patients in terms of encryption computation, ciphertext storage, and energy consumption. In our future work, we will consider emergency conditions during data sharing and provide efficient access policy update and revocation.

REFERENCES

[1] Moreno Ambrosin, Mauro Conti, and Tooska Dargahi. 2015. On the feasibility of attribute-based encryption on smartphone devices. In *Proceedings of the Annual Conference on IoT Challenges in Mobile and Industrial Systems (IoT-Sys'15).* 49–54.

[2] Diego F. Aranha and Conrado Porto Lopes Gouvea. 2013. RELIC. Retrieved May 2, 2018 from https://github.com/relic-toolkit/relic.

[3] Joakim Borgh, Edith Ngai, Börje Ohlman, and AdeelMohammad Malik. 2017. Employing attribute-based encryption in systems with resource constrained devices in an information-centric networking context. In *Proceedings of the Global IoT Summit (GIoTS'17).* 1–6.

[4] Raphael Bost, RalucaAda Popa, Stephen Tu, and Shafi Goldwasser. 2015. Machine learning classification over encrypted data. In *Proceedings of the Annual Network and Distributed System Security Symposium (NDSS'15).* 1–14.

[5] Yu Cao, Peng Hou, Donald Brown, Jie Wang, and Songqing Chen. 2015. Distributed analytics and edge intelligence: Pervasive health monitoring at the era of fog computing. In *Proceedings of the Annual Conference on Mobidata.* 43–48.

[6] Min Chen, Yongfeng Qian, Jing Chen, Kai Hwang, Shiwen Mao, and Long Hu. 2016. Privacy protection and intrusion avoidance for cloudlet-based medical data sharing. *IEEE Trans. Cloud Comput.* (2016). DOI: 10.1109/TCC.2016.2617382

[7] ChengKang Chu, ShermanSM Chow, WenGuey Tzeng, Jianying Zhou, and RobertH Deng. 2014. Key-aggregate cryptosystem for scalable data sharing in cloud storage. *IEEE Trans. Parallel Distrib. Syst.* 25, 2 (2014), 468–477.

[8] Dan Dobre, Paolo Viotti, and Marko Vukolić. 2014. Hybris: Robust hybrid cloud storage. In *Proceedings of the ACM Symposium on Cloud Computing (SoCC'14).* 1–14.

[9] Martin Dugas, Philipp Neuhaus, Alexandra Meidt, Justin Doods, Michael Storck, Philipp Bruland, and Julian Varghese. 2016. Portal of medical data models: Information infrastructure for medical research and healthcare. *Database: The Journal of Biological Databases & Curation (Oxford)* 2016, Article bav121 (2016). DOI: 10.1093/database/bav121

[10] Yaniv Harel, Irad Ben Gal, and Yuval Elovici. 2017. Cyber security and the role of intelligent systems in addressing its challenges. *ACM Trans. Intell. Syst. Technol.* 8, 4 (2017), 49.

[11] Cheng Huang, Rongxing Lu, Hui Zhu, Jun Shao, and Xiaodong Lin. 2016. FSSR: Fine-grained EHRs sharing via similarity-based recommendation in cloud-assisted eHealthcare system. In *Proceedings of the Annual Conference of the ACM ASIA Conference on Computer and Communications Security (AisaCCS'16).* 95–106.

[12] Jiawen Kang, Rong Yu, Xumin Huang, Maoqiang Wu, Sabita Maharjan, Shengli Xie, and Yan Zhang. 2018. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. *IEEE IoT J.* 6, 3 (2018), 4660–4670. DOI: 10.1109/JIOT.2018.2875542

[13] Eduard Kovacs. 2013. FDA Issues Alert Over Vulnerable Hospira Drug Pumps. Retrieved May 2, 2018 from http://www.securityweek.com/fda-issues-alert-over-vulnerable-hospira-drug-pumps.

[14] Jin Li, YanKit Li, Xiaofeng Chen, PatrickPC Lee, and Wenjing Lou. 2015. A hybrid cloud approach for secure authorized deduplication. *IEEE Trans. Parallel Distrib. Syst.* 26, 5 (2015), 1206–1216.

[15] Jin Li, Yinghui Zhang, Xiaofeng Chen, and Yang Xiang. 2018. Secure attribute-based data sharing for resource-limited users in cloud computing. *Comput. Secur.* 72 (2018), 1–12.

[16] Ming Li, Shucheng Yu, Yao Zheng, Kui Ren, and Wenjing Lou. 2013. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *IEEE Trans. Parallel Distrib. Syst.* 24, 1 (2013), 131–143.

[17] Qing Liu, BryanP Yan, CheukMan Yu, YuanTing Zhang, and CarmenCY Poon. 2014. Attenuation of systolic blood pressure and pulse transit time hysteresis during exercise and recovery in cardiovascular patients. *IEEE Trans. Biomed. Eng.* 61, 2 (2014), 346–352.

[18] W Liu and E. K. Park. 2014. Big data as an e-health service. In *Proceedings of the International Conference on Computing, Networking and Communication (ICNC'14).* 982–988.

[19] Ximeng Liu, Robert H. Deng, Yang Yang, Hieu N. Tran, and Shangping Zhong. Hybrid privacy-preserving clinical decision support system in fog–cloud computing. (unpublished).

[20] Ximeng Liu, Rongxing Lu, Jianfeng Ma, Le Chen, and Baodong Qin. 2016. Privacy-preserving patient-centric clinical decision support system on naive Bayesian classification. *IEEE J. Biomed. Health Inf.* 20, 2 (2016), 655–668.

[21] Yi Liu, Yinghui Zhang, Jie Ling, and Zhusong Liu. 2017. Secure and fine-grained access control on e-healthcare records in mobile cloud computing. *Fut. Gener. Comput. Syst.* 78, 3 (2017).

[22] Xuhong Peng, Ju Ren, Liang She, Deyu Zhang, Jie Li, and Yaoxue Zhang. 2018. Boat: A block-streaming app execution scheme for lightweight iot devices. *IEEE IoT J.* 5, 3 (2018), 1816–1829.

[23] Aarathi Prasad, Xiaohui Liang, and David Kotz. 2014. Poster: Balancing disclosure and utility of personal information. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys'14)*. 380–381.

[24] Amir Rahmani, Tuan Gia, Behailu Negash, Arman Anzanpour, Iman Azimi, Mingzhe Jiang, and Pasi Liljeberg. 2017. Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. *Fut. Gener. Comput. Syst.* 78 (2017), 641–658.

[25] Ju Ren, Hui Guo, Chugui Xu, and Yaoxue Zhang. 2017. Serving at the edge: A scalable IoT architecture based on transparent computing. *IEEE Netw.* 31, 5 (2017), 96–105.

[26] Adi Shamir. 1979. How to share a secret. *Commun. ACM* 22, 11 (1979), 612–613.

[27] Jian Shen, Tianqi Zhou, Xiaofeng Chen, Jin Li, and Willy Susilo. 2017. Anonymous and traceable group data sharing in cloud computing. *IEEE Trans. Inf. Forens. Secur.* 13, 4 (2017), 912–925.

[28] Wenjuan Tang, Kuan Zhang, Ju Ren, Yaoxue Zhang, and Xuemin Shen. 2019. Flexible and efficient authenticated key agreement scheme for bans based on physiological features. *IEEE Trans. Mobile Comput.* 18, 4 (2019), 845–856.

[29] Wenjuan Tang, Kuan Zhang, Deyu Zhang, Ju Ren, Yaoxue Zhang, and Xuemin Sherman Shen. 2019. Fog-enabled smart health: Toward cooperative and secure healthcare service provision. *IEEE Commun. Mag.* 57, 5 (2019), 42–48.

[30] Yue Tong, Jinyuan Sun, Sherman Chow, and Pan Li. 2013. Towards auditable cloud-assisted access of encrypted health data. In *Proceedings of the Annual Conference of the Canadian Nuclear Society*. 514–519.

[31] Shulan Wang, Junwei Zhou, Joseph K. Liu, Jianping Yu, Jianyong Chen, and Weixin Xie. 2016. An efficient file hierarchy attribute-based encryption scheme in cloud computing. *IEEE Trans. Inf. Forens. Secur.* 11, 6 (2016), 1265–1277.

[32] Xinlei Wang, Jianqing Zhang, Eve M. Schooler, and Mihaela Ion. 2014. Performance evaluation of attribute-based encryption: Toward data privacy in the IoT. In *Proceedings of the Annual Conference on IEEE International Conference on Communications (ICC'14)*. 725–730.

[33] Brent Waters. 2011. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Proceedings of the International Conference on Theory and Practice in Public Key Cryptography (PKC'11)*. 53–70.

[34] Chugui Xu, Ju Ren, Yaoxue Zhang, Zhan Qin, and Kui Ren. 2017. Dppro: Differentially private high-dimensional data release via random projection. *IEEE Trans. Inf. Forens. Secur.* 12, 12 (2017), 3081–3093.

[35] Yang Xu, Ju Ren, Guojun Wang, Cheng Zhang, Jidian Yang, and Yaoxue Zhang. 2019. A blockchain-based nonrepudiation network computing service scheme for industrial IoT. *IEEE Trans. Industr. Inf.* 15, 6 (2019), 3632–3641. DOI: 10.1109/TII.2019.2897133

[36] Zhongyuan Xu and ScottD Stoller. 2015. Mining attribute-based access control policies. *IEEE Trans. Depend. Sec. Comput.* 12, 5 (2015), 533–545.

[37] JiJiang Yang, JianQiang Li, and Yu Niu. 2015. A hybrid solution for privacy preserving medical data sharing in the cloud environment. *Fut. Gener. Comput. Syst.* 43 (2015), 74–86.

[38] Kan Yang, Zhen Liu, Xiaohua Jia, and Xuemin Shen. 2016. Time-domain attribute-based access control for cloud-based video content sharing: A cryptographic approach. *IEEE Trans. Multimed.* 18, 5 (2016), 940–950.

[39] LoYao Yeh, WoeiJiunn Tsaur, and HsinHan Huang. 2017. Secure IoT-based, incentive-aware emergency personnel dispatching scheme with weighted fine-grained access control. *ACM Trans. Intell. Syst. Technol.* 9, 1 (2017), 10.

[40] Hui Yin, Zheng Qin, Lu Ou, and Keqin Li. 2017. A query privacy-enhanced and secure search scheme over encrypted data in cloud computing. *J. Comput. Syst. Sci.* 90 (2017), 14–27.

[41] Kuan Zhang, Xiaohui Liang, Jianbing Ni, Kan Yang, and Xuemin Shen. Exploiting social network to enhance human-to-human infection analysis without privacy leakage. (unpublished).

[42] Kuan Zhang, Kan Yang, Xiaohui Liang, Zhou Su, Xuemin Shen, and Henry H. Luo. 2015. Security and privacy for mobile healthcare networks: From a quality of protection perspective. *IEEE Wireless Commun.* 22, 4 (2015), 104–112.

[43] Lide Zhang, Birjodh Tiwana, Zhiyun Qian, Zhaoguang Wang, Robert P. Dick, ZhuoqingMorley Mao, and Lei Yang. 2010. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proceedings of the Annual Conference on International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS'10)*. 105–114.

[44] Shaobo Zhang, Xiong Li, Zhiyuan Tan, Tao Peng, and Guojun Wang. 2019. A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services. *Fut. Gener. Comput. Syst.* 94 (2019), 40–50.

[45] Yin Zhang, Meikang Qiu, ChunWei Tsai, MohammadMehedi Hassan, and Atif Alamri. 2017. Health-CPS: Healthcare cyber-physical system assisted by cloud and big data. *IEEE Syst. J.* 11, 1 (2017), 88–95.

[46] Alicia L. Nobles, Ketki Vilankar, Hao Wu, and Laura E. Barnes. 2015. Evaluation of data quality of multisite electronic health record data for secondary analysis. In *Proceedings of IEEE International Conference on Big Data (BigData'15)*. 2612–2620.

Received March 2019; revised May 2019; accepted June 2019