This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2916569, IEEE Transactions on Dependable and Secure Computing

Secure Fine-Grained Encrypted Keyword Search for E-Healthcare Cloud

Haijiang Wang, Jianting Ning, Xinyi Huang, Guiyi Wei, Geong Sen Poh, and Ximeng Liu, *Member, IEEE*

Abstract—E-Healthcare systems are increasingly popular due to the introduction of wearable healthcare devices and sensors. Personal health records (PHRs) are collected by these devices and stored in a remote cloud. Due to privacy concern, these records should not be accessible by any unauthorized party, and the cloud providers should not be able to learn any information from the stored records. To address the above issues, one promising solution is to employ attribute based encryption (ABE) for fine-grained access control and searchable encryption for keyword search on encrypted data. However, most of existing ABE schemes leak the privacy of access policy which may also contain sensitive information. On the other hand, for users' devices with limited computing power and bandwidth, the mechanism should enable them to be able to search the PHRs efficiently. Unfortunately, most existing works on ABE do not support efficient keyword search on encrypted data. In this work, we propose an efficient hidden policy ABE scheme with keyword search. Our scheme enables efficient keyword search with constant computational overhead and constant storage overhead. Moreover, we enhance the recipient's privacy which hides the access policy. As of independent interest, we present a trapdoor malleability attack and demonstrate that some of previous schemes may suffer from such attack.

Index Terms— E-Healthcare Cloud, Attribute-Based Encryption, Hidden Policy, Searchable Encryption, Keyword Search

1 INTRODUCTION

1.1 Background

In recent years, with the development of Internet of Things and the popularization of mobile communication equipment, electronic medicine has been highly valued and developed rapidly. As shown in Fig. 1, by using wireless sensor technology, patients' heart rate, blood pressure, breathing, electrocardiogram and other data can be collected in an all-round way. With M2M technology, through the existing communication network, the remote patient sensitive information can be transmitted to various medical institutions, doctors or other medical service personnel, which can facilitate patients to understand body indicators and other data at any time and anywhere, so that the therapeutic effect and drug use can be effectively monitored. With these benefits, more and more patients outsource their data to the cloud server and later access them with mobile devices from anywhere. However, a key concern is data privacy especially for highly sensitive information, such as personal health records (PHRs) collected from resource-constrained wearable healthcare devices and sensors. In these records, patients' privacy information includes basic identity information, case report and inspection report. These privacy information, if stored in plaintext, is vulnerable to attacks by malicious users who can easily access the private information of other users. Under this environment, the cloud server or malicious patients may be motivated to access and derive the sensitive information. To achieve privacy, the PHRs should be stored in the cloud with encrypted form. However, it can be difficult for patients or doctors to retrieve certain data due to encryption, since with the encrypted PHRs, keyword search functionality becomes an especially challenging issue [44].

A trivial solution is that patients or doctors download and decrypt the entire encrypted records locally and then search for the desired results from the plaintext data. In contrast to this inefficient method, searchable encryption (SE) has been proposed as a better solution. In a searchable encryption system, data user stores the encrypted private data in the cloud server. In order to conduct keyword search, a user constructs a search trapdoor and submits it to the server. With the search trapdoor, the server can search over the encrypted data. The search trapdoor is generated by the user's private key and some keywords. Moreover, the search trapdoor can only be constructed by users with search privileges and search private key. Informally, searchable encryption enables user to search over the encrypted data without decryption. Solutions for private keyword search have been proposed in the literature [10], [20], [45], [49], [56]. Unfortunately, all these schemes do not support fine-grained access control. However, for the e-healthcare cloud system to be effective, it must support access control by multiple patients (data owners) and multiple doctors (data users). Furthermore, it must enable multiple doctors/patients with resource-constrained devices to query over the encrypted PHRs generated by multiple patients, and achieves fine-

- *Haijiang Wang and Guiyi Wei are with School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Zhejiang,* 200240*, China. E-mails: wanghaijiangyes@163.com, weigy@zjgsu.edu.cn.*
- *Jianting Ning is with Department of Computer Science, National University of Singapore, Singapore. E-mail: jtning88@gmail.com. (Corresponding author)*
- *Xinyi Huang is with the School of Mathematics and Computer Science, Fujian Normal University, China. E-mail: xyhuang81@gmail.com.*
- *Geong Sen Poh is with NUS-Singtel Cyber Security R & D Lab, Singapore. E-mail: geongsen@gmail.com.*
- *Ximeng Liu is with the School of Information Systems, Singapore Management University, Singapore, and College of Mathematics and Computer Science, Fuzhou University, China. E-mail: snbnix@gmail.com.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2916569, IEEE Transactions on Dependable and Secure Computing

grained keyword search authorization over the encrypted PHRs.

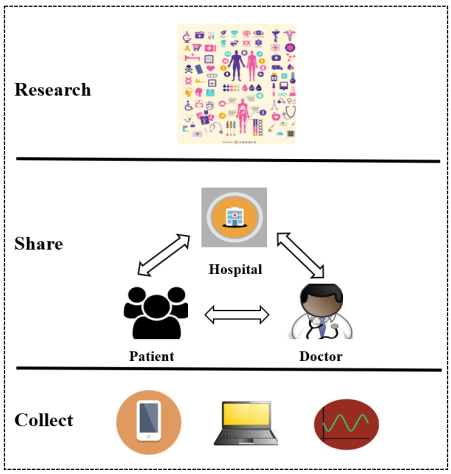

Fig. 1: E-Healthcare System

A cryptographic solution to fulfill the above requirements is attribute based encryption (ABE). ABE technology has been used to design access control system, which can be deployed for fine-grained access control in e-healthcare cloud computing system. Inspired by ABE, Sun *et al.* [40] and Zheng *et al.* [54] independently proposed attribute based searchable encryption (ABSE) schemes. ABSE schemes can be treated as an integration of ABE and SE. In the case for e-healthcare system, using ciphertext policy attribute based encryption (CP-ABE), patients may decide on an access policy that qualifies a doctor's ability not only for decryption but also for keyword search. Only doctors who own sufficient attributes to satisfy that policy can generate a valid trapdoor, which helps the cloud server search over the encrypted PHRs. Solutions for both key-policy ABE [16], [17], [23], [28] and ciphertext-policy ABE [1], [29], [33] have been proposed. Unfortunately, all these schemes do not support policy hiding. In attribute-based encryption system, policies are used to specify the access rights of users. In most attribute-based encryption schemes, in order to achieve decryption operation, user usually needs to explicitly attach the policy to the ciphertext. However, generally, the policy may contain some sensitive information. If the policy is leaked, cloud server or malicious users can infer the user's sensitive information through the leaked policy. Hence, a secure attribute-based encryption scheme should provide access policy hiding. With the implementation of policy hiding mechanism, the patient does not need to attach the access policy to the ciphertext in plaintext form, thus avoiding the privacy information leaked directly by the access policy, thus improving the security of the whole system.

Moreover, in the case of patients with resource-constrained devices, an ideal SE solution should be with constant communication and computational cost. Although existing ABSE schemes provide fine-grained access control with keyword search, it suffers from huge computation cost and huge communication cost. In ABSE schemes, the length of ciphertext and the length of secret keys increase with the number of attributes involved in the policy. A large number of ciphertext and keys make the system have a large computational and communication overhead. At the same time, in the decryption stage, a large number of pairing operations incurs huge computational cost. Unfortunately, this large amount of computing requires users to have huge computing power, which is contrary to the lightweight design concept in cloud computing services.

1.2 Our Contributions and Results

The policy leakage and computationally expensive search problems, as described above, are the main obstacles in using attribute based encryption system in our e-healthcare scenario. In this work, we propose a new secure fine-grained encrypted keyword search scheme, termed *Fast Keyword Search-Hidden Policy ABE (FKS-HPABE)*, that addresses these problems. Our contributions are as follows:

1) *Fine-grained Keyword Search Authorization*. We propose an authorized keyword search scheme over the encrypted PHRs in the e-healthcare cloud computing system. Our system supports fine-grained search authorization, whereby only doctors who own sufficient attributes to satisfy the patient enforced policy can generate valid trapdoors and access the corresponding PHRs.
2) *Hidden Policy*. Our system not only guarantees privacy of PHRs, but also preserves the privacy of the access policy in the ciphertext, which provides recipient anonymity.
3) *Constant Overhead*. The system obtains the property of constant overhead as shown in Table 1. In our new system, the sizes of ciphertext and secret key are independent of number of attributes and number of the values of the attributes, which does not increase linearly with the number of values of the attributes.
4) *Fast Keyword Search*. In traditional ABSE schemes, all secret key components need pairings with the corresponding components in ciphertext, which make the search time proportional to the number of attributes used during searching. In our system, with the constant size ciphertext and constant size secret key, only two pairings are needed, so that the search process is faster than previous ABSE schemes.

1.3 Related Works

In principle, fully homomorphic encryption (FHE) [22] can evaluate any functions over encrypted data which can be used to design encrypted information retrieval in e-healthcare system. However, this construction requires heavy computation complexity [11]. To realize efficient encrypted keyword search, Song *et al.* [39] introduced the notion of symmetric searchable encryption, and Boneh *et al.* [3] proposed the first public key encryption with keyword search scheme. After that, a series of SE schemes were proposed [10], [18], [45] aiming at enrich the search functionality including result ranking [13], [20] verifiable-keyword search [21], [43] multi-keyword search [18], [49] and fuzzy keyword search [46], [51], [55]. Generally, searchable encryption mechanism provides keyword search functionality

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2916569, IEEE Transactions on Dependable and Secure Computing

TABLE 1: Performance comparison with other related works [1,2,3]

	PK	SK	CS
[36]	$1\|\mathbb{G}_T\| + (mn+3)\|\mathbb{G}\|$	$(2n+1)\|\mathbb{G}\| + 1\|\mathcal{Z}_P\|$	$2\|\mathbb{G}_T\| + (m+1)n\|\mathbb{G}\|$
[40]	$1\|\mathbb{G}_T\| + (3n+1)\|\mathbb{G}\|$	$(2n+1)\|\mathbb{G}\| + 2\|\mathcal{Z}_P\|$	$(n+2)\|\mathbb{G}\| + 1\|\mathcal{Z}_P\|$
[44]	$1\|\mathbb{G}_T\| + 2\|\mathbb{G}\|$	$(n+3)\|\mathbb{G}\|$	$1\|\mathbb{G}_T\| + (n+2)\|\mathbb{G}\|$
Our work	$5\|\mathbb{G}\|$	$1\|\mathcal{Z}_P\| + 2\|\mathbb{G}\|$	$3\|\mathbb{G}_T\| + 3\|\mathbb{G}\| + 1\|\mathcal{Z}_P\|$

[1] PK stands for Public Key Size, SK stands for Secret Key Size, CS stands for Ciphertext Size.
[2] Let $|\mathbb{G}|$ denote a bit length of an element of $\mathbb{G}$, $|\mathbb{G}_T|$ denote a bit length of an element of $\mathbb{G}_T$ and $|\mathcal{Z}_P|$ denote a bit length of an element of $\mathcal{Z}_P$.
[3] n denote the number of attributes in the access structure and m denote the number of possible values for an attribute.

over encrypted data without leaking any information about plaintext or the query. Unfortunately, most of the above schemes only support single user operate encrypted keyword search. However, in the e-healthcare system, there are many patients and many doctors. Hence, it is necessary for the e-healthcare system to provide encrypted information sharing for patients and doctors.

To realize encrypted information sharing, Li *et al.* [25] presented a framework for authorized private keyword search with proxy re-encryption. The authors employed hierarchical predicate encryption to construct a scheme for authorized private keyword search. In the proposed scheme, there are several local trusted authorities who are in charge of determining doctor' search privileges. Liang *et al.* [26] combined proxy re-encryption with anonymous technique to realize encrypted data conditionally shared multiple times without leaking both the message and the identity. Zhang *et al.* [52] proposed a fine-grained data access control scheme by exploiting the techniques of proxy re-encryption and fair watermarking. A key-disclosure and offline data owner authorized keyword search was proposed by Zhou *et al.* [56]. After that, a series of encrypted information sharing schemes were proposed. However, in the multi-user e-healthcare system, it is necessary to realize fine-grained access control for the encrypted information.

Sahai *et al.* [38] introduced the concept of ABE, that realizes fine-grained access control to protect encrypted sensitive information [31], [32]. There are two types of ABE schemes, namely key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). For KP-ABE scheme, each encrypted data is related to a set of attributes, and each doctor's secret key is associated with an access policy. A doctor is able to decrypt an encrypted data if and only if the attribute set related to the ciphertext satisfies the access policy associated with the doctors secret key. For CP-ABE scheme, the roles of an attribute set and an access policy are reversed. In recent years, a series of constructions for fine-grained access control of e-healthcare are proposed with (KP-ABE) [16], [17], [23], [28] and (CP-ABE) [1], [29], [33], which mainly concern the issue of data confidentiality, while leaving the challenging problems of realizing hidden policy and constant system cost.

Nevertheless, when implementing an ABE system, the size of the ciphertext is proportional to the number of attributes associated with it [5], [8], [14], [34], [41], [47]. Besides, only users who own sufficient attributes satisfy that policy can send valid search trapdoor to cloud server. So, all the secret key components are required in performing pairing operations with the corresponding components in ciphertext, which make the search time proportional to the number of attributes used during search. Specifically, one pairing operation per attribute is needed during search and decryption. Furthermore, in the e-healthcare cloud computing system, the access policy also contains some sensitive useful information of the data owner. For example, "illness" is not allowed to be leaked to an unauthorized user. However, in most of the existing schemes, the access policy is sent along with the ciphertext explicitly, which means these schemes do not provide recipient anonymity. ABE scheme that provides recipient anonymity via policy hiding while ensuring constant-size secret key, constant ciphertext length and constant computation cost with strong security level is a crucial problem. To solve this problem, ABE schemes with constant overhead were proposed [7], [37], [42]. In these schemes, both the sizes of ciphertext and secret key are at the constant level, which greatly reduce the computational and communication overhead of the system. Most importantly, in the decryption phase, the decrypting user only needs constant-level pairing to decrypt the ciphertext. This design conforms to the design concept of lightweight cryptography. Therefore, the implementation of constant-level ciphertext and constant-level key mechanism are conducive to improve the efficiency of a cryptosystem, which provides favorable conditions for the widespread popularity of cryptosystem.

However, all the above systems do not simultaneously support both encrypted keyword search and lightweight fine-grained access control in practice, which limit the commercial applications of ABE. Therefore, in this paper, we construct a constant overhead hidden policy ABE scheme with fast keyword search for a practical e-healthcare cloud systems.

1.4 Organization

Section 2 introduces the access structure of our scheme and some relevant preliminaries. Section 3 gives the formal definition of our hidden policy attribute based encryption with efficient keyword search system and its security model. Section 4 presents the construction of our FKS-HPABE system as well as the security proof. Section 5 gives the comparison between our scheme and some other related works. Section 6 gives our trapdoor malleability attack as well as the security proof of our scheme. Section 7 presents a briefly conclusion.

2 PRELIMINARIES

2.1 Access Structure

There are several kinds of access structures in ABE schemes, such as threshold structure [29], [44], tree-based structure

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2916569, IEEE Transactions on Dependable and Secure Computing

[1], [16], AND-gate structure [36], [40]. In our construction, we exploit a series of AND-gate on multi-value attributes as our access structure. Assume the total number of attributes are n, so that all n attributes can be represented as $U = \{att_1, att_2, ..., att_n\}$. For each attribute $att_i \in U, (i = 1, 2, ..., n)$, it can take values in set $V_i = \{v_{i,1}, v_{i,2}, ..., v_{i,n_i}\}$, where n_i is the number of the possible values for att_i. An attribute list S for a user can be represented as $S = (x_1, x_2, ..., x_n)$, where $x_i \in V_i$, and an access policy in ciphertext can be repetsented as $\mathbb{A} = (w_1, w_2, ..., w_n)$, where $w_i \in V_i$. We say that an attribute list S satisfies an access policy $\mathbb{A}$ if and only if $x_i = w_i, (i = 1, 2, ..., n)$.

2.2 Bilinear Map

We briefly review the necessary facts about bilinear maps and bilinear map groups (For more detail, see [2]). Consider the following setting:

- $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p.
- $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. The bilinear map e has the following properties:
  (1) Bilinearity: $\forall u, v \in \mathbb{G}$ and $a, b \in \mathcal{Z}_P$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
  (2) Non-degeneracy: $e(g, g) \neq 1$.

2.3 Complexity Assumption

The security of our protocol is based on the hardness of the truncated decision augmented bilinear Diffie-Hellman exponent assumption.

The decisional version of truncated n-ABDHE is defined as follow. Let $\mathbb{G}$ be bilinear group of prime order p and g, g' are two independent generators of $\mathbb{G}$. Given

$$(g', g'_{n+2}, g_1, ..., g_n) \in \mathbb{G}^{n+2}$$

where $g_i = g^{\alpha^i}$ for some unknown $\alpha \in \mathcal{Z}_P^*$. An algorithm $\mathcal{B}$ that outputs $b \in \{0, 1\}$ has advantage ε in solving the decision n-BDHE problem if

$$\begin{aligned} &| \, pr[\mathcal{B}(g', g'_{n+2}, g_1, ..., g_n, e(g_{n+1}, g')) = 0] \\ &\quad - pr[\mathcal{B}(g', g'_{n+2}, g_1, ..., g_n, Z) = 0] \, | \geq \varepsilon \end{aligned}$$

where the probability is over the random choice of g, g' in $\mathbb{G}$, the random choice $\alpha \in \mathbb{Z}_p^*$, the random choice of $Z \in \mathbb{G}_T$, and the random bits consumed by $\mathcal{B}$. We say that the truncated (τ, ε, n)-ABDHE assumption holds in $\mathbb{G}$ if no τ-time algorithm has advantage at least ε in solving the truncated n-ABDHE problem.

3 PROBLEM FORMULATION

3.1 System Model

In this section, we present the architecture of a searchable e-healthcare secure storage system suitable for lightweight devices as shown in Fig. 2. The system consists of four entities: patients, doctors (patients and doctors are collectively referred to as system users), trusted authority and the cloud server. The roles are defined as follows:

- **Trusted Authority** : The trusted authority is mainly responsible for system users' attribute management and the system users' secret key distribution. The trusted authority is located in the system as the core management organization, which is completely trusted by doctors and patients. This is a constraint of this system.
- **Patients** : Through wireless sensors the system realizes real-time collection, statistics and other functions of patient information. To protect data privacy, patients encrypt their PHRs with ABE technology: In the encryption phase, patients encrypt the records with an access policy. The specified access policy is used to indicate the group of doctors whom can access and decrypt these records.
- **Doctors** : The patient's personal health information is encrypted by the personal device with ABE technology, and then stored in the cloud server. Only the authorized doctors have access control authority and specifies the corresponding diagnosis and treatment plan. In order to achieve privacy-preserving information retrieval, doctors can use their own secret keys to generate secure search trapdoors to perform keyword search over the encrypted PHRs.
- **Cloud Server** : The cloud server is responsible for health records storage and keyword search operations. When a doctor asks for a search request with a search trapdoor, the server matches the search trapdoor one by one with the encrypted PHRs. If and only if the keyword contained in the search trapdoor is the same as the keyword contained in the ciphertext, and the attribute set associated with the secret key used to generate search trapdoor meet the access policy specified in the ciphertext, the cloud server returns the matched result to the doctor who performed the query.

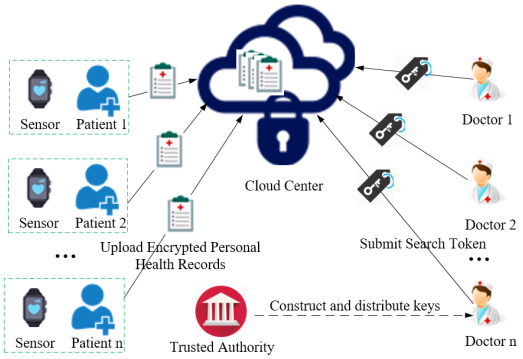

Fig. 2: System Model

3.2 Threat Model

We assume that the cloud server is "honest-but-curious", which follows many related works on e-healthcare cloud computing system [5], [8], [14], [41], [45], [47]. The cloud server performs protocol algorithms honestly, however, may exploits some threats to attack the system as follows:

1) The could server "honestly" follows the designated protocol, but "curiously" learns information based on the data available to him, and infers addition-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2916569, IEEE Transactions on Dependable and Secure Computing

al privacy information of the encrypted personal healthcare records content or the search query.
2) The cloud server may collude with any number of doctors to derive additional information about other doctors' query privacy.
3) The unauthorized doctors may collude to access data beyond their access privileges.

3.3 Design Goals

The system design of FKS-HPABE over encrypted PHRs in e-healthcare cloud system should achieve the following main security and performance goals:

- **Fine-grained Keyword Search Authorization:** The secure system should enable patients to enforce search authorization, i.e. only doctors with sufficient attributes satisfying that policy can generate valid search trapdoor and obtain valid search results.
- **Scalability and Efficiency:** The system should allow multiple patients to encrypt data, while at the same time enable a group of doctors to perform keyword search. Moreover, the designed system should achieve constant computation and communication cost based on the limited computing power and bandwidth setting.
- **Security Goals:** The primary security goals are to prevent the cloud server and other malicious users from learning any useful information about the encrypted data, index, and the users' query, except what can be derived from the search results. We define them as follow. 1) *Semantic Security:* As an ABE encryption scheme, we will formally prove our scheme is semantic secure against *chosen keyword and plaintext attack* under *fully ciphertext policy model* (IND-CKA-CPA). 2) *Trapdoor unmalleability:* This security property makes the malicious user unable to deduce a valid trapdoor of keyword w' when given a valid trapdoor of keyword w.

3.4 Security Definition for FKS-HPABE

We define security for attribute based encryption with fast keyword search system in the sense of semantic security. Formally, we define security against an active adversary $\mathcal{A}$ using the following game between a challenger $\mathcal{B}$ and the adversary $\mathcal{A}$.

- **Setup:** Challenger $\mathcal{B}$ runs the **Setup** algorithm and forwards public parameters pp to adversary $\mathcal{A}$.
- **Phase 1:** Adversary $\mathcal{A}$ can adaptively make secret key queries and trapdoor queries as follows:
  1) Secret key query: Adversary $\mathcal{A}$ can adaptively ask the challenger $\mathcal{B}$ for the secret key for sets of $S_1, S_2, ..., S_{q_1}$.
  2) Trapdoor query: Adversary $\mathcal{A}$ can adaptively ask the challenger $\mathcal{B}$ for the trapdoor for the keyword w with the attribute sets $S_1, S_2, ..., S_{q_1}$.
- **Challenge:** At some point, $\mathcal{A}$ sends the challenger $\mathcal{B}$ two keywords w_0, w_1 and two messages M_0 and M_1 on which it wishes to be challenged. In addition the adversary $\mathcal{A}$ gives a challenge access structure $\mathbb{A}^*$. The only restriction is that the adversary $\mathcal{A}$ did not previously ask for the trapdoors t_{w_0},t_{w_1}, with the attribute set S_i which satisfy the challenge access structure $\mathbb{A}^*$. The challenger $\mathcal{B}$ picks two random bits $b, c \in \{0, 1\}$, encrypts M_b with w_c and gives adversary the challenge ciphertext $ct^*_{b,c}$ and $I^*_{w_{b,c}}$.
- **Phase 2:** $\mathcal{A}$ can continue to ask for secret key or trapdoors.
- **Guess:** Eventually, the adversary $\mathcal{A}$ outputs $b', c' \in \{0, 1\}$ and wins the game if $b = b'$ and $c = c'$.

The advantage of an adversary is defined to be $Adv = |\Pr[b = b' \wedge c = c'] - 1/4|$ in this game.

Definition 1. *(IND-CKA-CPA) An attribute based encryption with keyword search system is semantically secure against an adaptive chosen keyword attack and an adaptive chosen plaintext attack if all probabilistic polynomial-time (PPT) attackers have at most negligible advantage in λ in the above security game.*

4 THE FKS-HPABE SYSTEM FOR MOBILE E-HEALTHCARE CLOUD

4.1 Definition

An attribute based searchable encryption system consists of six algorithms: Setup, KeyGen, Encrypt, Trapdoor, Search and Decrypt. The specific execution description of these algorithms are as follows:

- **Setup**$(1^\lambda) \rightarrow (pp, msk)$: The trusted authority executes **Setup** algorithm to initialize the whole system. Input security parameter λ, the algorithm outputs public parameters pp and master secret key msk. The trusted authority keep the master secret key confidential, which is used to generate users' secret keys. After that, the trusted authority publishes the public parameters.
- **KeyGen**$(msk, S) \rightarrow sk$: The trusted authority executes **KeyGen** algorithm to generate secret key for each user. Input master secret key msk and a set of attributes S, the algorithm outputs secret key sk associated with S.
- **Encrypt**$(pp, m, w, \mathbb{A}) \rightarrow (ct, I_w)$: The patient executes **Encrypt** algorithm to generate searchable ciphertext stored on the cloud server. Input public parameters pp, message plaintext m, the extracted keyword w and the specified access policy $\mathbb{A}$, the algorithm produces ciphertext ct and keyword index I_w such that a doctor's secret key associated with attribute set S can be used to search and decrypt ciphertext if and only if $S \models \mathbb{A}$.
- **Trapdoor**$(sk, w) \rightarrow t_w$: The query doctor executes **Trapdoor** algorithm to generate secure search trapdoors to help cloud server to search over the encrypted PHRs. Input secret key sk and a keyword w, the algorithm outputs a trapdoor t_w.
- **Search**$(I_w, t_w) \rightarrow 0$ *or* 1 : The cloud server executes **Search** algorithm to check whether a given index ciphertext I_w contains the keyword w contained in trapdoor t_w. Input index ciphertext I_w and search

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2916569, IEEE Transactions on Dependable and Secure Computing

trapdoor t_w, the algorithm outputs 0 if the index ciphertext and trapdoor contain the same keyword; else, outputs 1, and then delivers the corresponding message ciphertext to the query doctor.

- **Decrypt**$(sk, ct) \rightarrow m$: The query doctor executes **Decrypt** algorithm to recover the message plaintext. Input ciphertext ct and secret key sk, the algorithm return message m if and only if the attribute set S associated with secret key sk satisfies the access policy $\mathbb{A}$ contained in the ciphertext.

4.2 System Flow

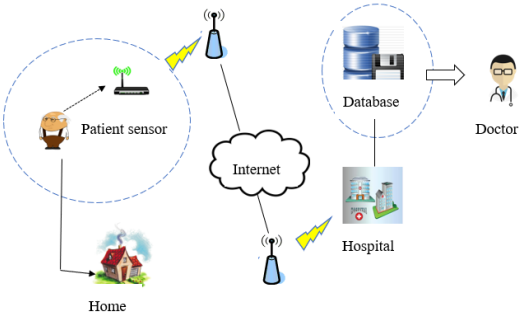

Fig. 3: System Flow

In this subsection, we describe a concrete application of our FKS-HPABE scheme in e-healthcare system. As shown in Fig. 3, based on the system model defined in Section 3, our FKS-HPABE scheme works as follows:

1) In step 1, the mobile medical sensors of patients can collect real-time healthcare information such as blood pressure and heart rate at home. In order to protect patients' privacy, these information is encrypted with CP-ABE and then uploaded to the network. Specifically, patients can specify an access policy to designate doctors who can search and decrypt the private data.
2) In step 2, with M2M technology, through the existing communication network, the patient healthcare information can be transmitted to various medical institutions such as hospitals or medical service research centers.
3) In step 3, doctors gain the secret keys generated by the trusted authority. With the secret key, a doctor can authorize encrypted keyword search operation to the cloud server by submitting a secure trapdoor.
4) In step 4, with the trapdoor, the cloud server is able to perform keyword search over the encrypted personal health records. If and only if the attribute set embedded in the secret key satisfies the specified access policy embedded in the ciphertext, the doctor can gain the relevant search results.

4.3 Our Technique

To realize hidden policy construction, we adopt the "anonymous" technique based on strong-RSA based signatures [15]. A private key for attribute list $S = (x_1, x_2, \cdots, x_n)$ is a tuple $(r_{ID}, h_{ID_1}, h_{ID_2})$, where $r_{ID} \in \mathbb{Z}_p$, $h_{ID_1} = (h_1 g^{-r_{ID}})^{1/\alpha - \Sigma_{i=1}^{n} H_1(x_i)}$ and $h_{ID_2} = (h_2 g^{-r_{ID}})^{1/\beta - \Sigma_{i=1}^{n} H_1(x_i)}$; In order to protect against collusion attack, different users with same attribute list have their private key components generated from the different r_{ID}. One can view the private key generation procedure as a strongly existentially unforgeable signature scheme under the q-strong DH assumption: that it is hard to compute a pair $(c, g^{1/(a-c)})$ given $\{g^{a^i} : i \in [0, q]\}$. Let $S = (x_1, x_2, \cdots, x_n)$ be an attribute list of a user, and $\mathbb{A} = (w_1, w_2, \cdots, w_n)$ be an access structure. If and only if $x_i = w_i, (i = 1, 2, ..., n)$, we say the attribute list satisfies the access structure. There is only one value for an attribute in the user's attribute list and the same with the access structure. So, there is no need to include an access structure in ciphertext, then hidden policy is achieved.

To realize fast keyword search, we adopt the "aggregation" technique from [12]. For user with the attribute list $S = (x_1, x_2, \cdots, x_n)$, key generation algorithm generates a private key as $(r_{ID}, h_{ID_1}, h_{ID_2})$, where $r_{ID} \in \mathbb{Z}_p$, $h_{ID_1} = (h_1 g^{-r_{ID}})^{1/\alpha - \Sigma_{i=1}^{n} H_1(x_i)}$ and $h_{ID_2} = (h_2 g^{-r_{ID}})^{1/\beta - \Sigma_{i=1}^{n} H_1(x_i)}$. User generates a trapdoor for keyword w with (r_{ID}, h_{ID_2}), and sends to cloud server. With the trapdoor, cloud server can search over the encrypted data. By introducing the "aggregation" technique, the search phase only needs two pairings, this is a great advantage over previous schemes, where the search time is proportional to the number of attributes used during search, then fast keyword search and fine-grained access control are achieved.

To realize resist trapdoor malleability attack, given secret key $(a \cdot r_{ID}, h_{ID})$ and keyword w, we generate a trapdoor as $t_w = \langle tk_1 = (g \cdot h^{H(w)})^s, tk_2 = \tilde{h}^s, tk_3 = a \cdot r_{ID} \cdot s, tk_4 = h_{ID}^s \rangle$. Based on the Discrete Logarithm Problem, given $tk_1 = (g \cdot h^{H(w)})^s, tk_2 = \tilde{h}^s$ and $H(w)$, the malicious user can not get s. Hence, given a valid trapdoor t_w of w, the malicious user cannot generate a valid trapdoor $t_{w'}$ of a different keyword w', thus our scheme protect the authorized user against trapdoor malleability attack.

The basic idea underlying our resistance against trapdoor malleability attack in ABSE scheme is to separate the index and search trapdoor into two parts: one is associated with the keyword and the other is associated with the access policy. If the data user's attributes satisfy the access control policy, it can determine whether the search trapdoor matches the encrypted keyword.

4.4 Our Construction

Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order p, and let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be the bilinear map. Our FKS-HPABE system works as follows:

- **Setup**(1^λ) : Randomly chooses $\alpha, \beta \in \mathbb{Z}_p$, $g, h_1, h_2 \in \mathbb{G}$, and three hash functions $H_1, H_2, H_3 : \{0,1\}^* \rightarrow \{0,1\}^{logp}$. It sets $g_1 = g^\alpha, g_2 = g^\beta$. The public parameters pp and master secret key msk are given by

$$pp = (g, g_1, g_2, h_1, h_2), \quad msk = (\alpha, \beta)$$

- **KeyGen**(msk, S) : Input master secret key msk and a set of attributes $S = (x_1, x_2, \cdots, x_n)$ and output the secret key sk. The key generation algorith-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2916569, IEEE Transactions on Dependable and Secure Computing

m selects random $r_{ID} \in \mathcal{Z}_P$, sets the secret key $sk = (r_{ID}, h_{ID_1}, h_{ID_2})$, where

$$h_{ID_1} = (h_1 g^{-r_{ID}})^{\frac{1}{\alpha - \Sigma_{i=1}^{n} H_1(x_i)}},$$
$$h_{ID_2} = (h_2 g^{-r_{ID}})^{\frac{1}{\beta - \Sigma_{i=1}^{n} H_1(x_i)}}$$

- **Encrypt**$(1^\lambda, pp, M, w, \mathbb{A})$: Given a message $M \in \mathbb{G}$ and an AND gate $\mathbb{A} = (w_1, w_2, \cdots, w_n)$, and the corresponding keyword w, the encryption algorithm selects random number $s \in \mathbb{Z}_p^*$ and sets the ciphertext ct as

$$\langle C_1 = g_1^s \cdot g^{-s \cdot \Sigma_{i=1}^{n} H_1(w_i)}, C_2 = e(g,g)^s,$$
$$C_3 = M \cdot e(g, h_1)^{-s}, v = H_3(C_1, C_2, C_3),$$
$$C_4 = g_2^v \cdot g^{-v \cdot \Sigma_{i=1}^{n} H_1(w_i)}, C_5 = e(g,g)^v,$$
$$C_6 = g^{v \cdot H_2(w)} \rangle$$

- **Trapdoor**(sk, w) : User U_i generates the trapdoor of his chosen keyword w as: selects random number $t \in \mathcal{Z}_P^*$ and sets the trapdoor t_w as

$$t_w = (td_1 = h_{ID_2}^{t \cdot H_2(w)}, td_2 = r_{ID} \cdot t \cdot H_2(w), td_3 = h_2^t)$$

- **Search**(ct, t_w) : Given the trapdoor $t_w = [td_1, td_2, td_3]$ and ciphertext ct. The cloud server checks the following equation

$$e(C_4, td_1) \cdot C_5^{td_2} = e(C_6, td_3)$$

If the equation holds return 1 and 0 otherwise.

- **Decrypt**(sk, ct) : If the attribute list S satisfies the access policy $\mathbb{A}$, i.e., $x_i = w_i, (i = 1, 2, ..., n)$. On input ciphertext ct and the secret key sk, output a message M as:

$$M = C_3 \cdot e(C_1, h_{ID_1}) \cdot C_2^{r_{ID}}$$

Correctness (Search): We now show the correctness of our **Search** phase : If the attribute list S satisfies the access policy $\mathbb{A}$, i.e., $a_i = b_i, (i = 1, 2, ..., n)$. The cloud server check whether a given ciphertext I_w contains the keyword w specified by the trapdoor t_w.

$$\begin{aligned} & e(C_4, td_1) \cdot C_5^{td_2} \\ &= e(g_2^v \cdot g^{-v \cdot \Sigma_{i=1}^{n} H_1(b_i)}, h_{ID_2}^{t \cdot H_2(w)}) \cdot e(g,g)^{v \cdot td_2} \\ &= e(g, h_2)^{v \cdot t \cdot H_2(w)} \cdot e(g,g)^{v \cdot t \cdot H_2(w) \cdot r_{ID}} \\ &\cdot e(g,g)^{-v \cdot t \cdot H_2(w) \cdot r_{ID}} \\ &= e(g, h_2)^{v \cdot t \cdot H_2(w)} = e(C_6, td_3) \end{aligned}$$

Correctness (Decrypt): We now show the correctness of **Dec** of our construction: If the attribute list S satisfies the access policy $\mathbb{A}$, i.e., $a_i = b_i, (i = 1, 2, ..., n)$:

$$\begin{aligned} & C_3 \cdot e(C_1, h_{ID_1}) \cdot C_2^{r_{ID}} \\ &= e(g_1^s \cdot g^{-s \cdot \Sigma_{i=1}^{n} H_1(b_i)}, (h_1 g^{-r_{ID}})^{\frac{1}{\alpha - \Sigma_{i=1}^{n} H_1(a_i)}}) \\ &\cdot C_3 \cdot C_2^{r_{ID}} \\ &= C_3 \cdot e(g, h_1)^s \cdot e(g,g)^{-s \cdot r_{ID}} \cdot e(g,g)^{s \cdot r_{ID}} \\ &= m \cdot e(g, h_1)^{-s} \cdot e(g, h_1)^s = m \end{aligned}$$

4.5 Security Analysis

In this subsection, we will give the security analysis for our proposed system.

Theorem 1. *Let $q = q_{ID} + 1$. Assume the truncated decision (τ, ϵ, q)-ABDHE assumption holds for $(\mathbb{G}, \mathbb{G}_T, e)$. Then, the above ABKS system is $(\tau', \epsilon', q_{ID})$ IND-CKA-CPA secure for $\tau' = \tau - O(t_{exp} \cdot q^2)$ and $\epsilon' = \epsilon + 2/p$, where t_{exp} is the time required to exponentiate in $\mathbb{G}$.*

Proof. We now prove that our FKS-HPABE system is IND-CKA-CPA secure under the truncated decision (q_{ID}+1)-ABDHE assumption. Let $\mathcal{A}$ be an adversary that $(\tau', \epsilon', q_{ID})$-breaks the IND-CKA-CPA security of our ABKS system. We construct a simulator $\mathcal{B}$ that solves the truncated decision q-ABDHE problem. $\mathcal{B}$ interacts with $\mathcal{A}$ in the security game as follows:

- **Setup:** The challenger $\mathcal{B}$ takes as input a random truncated decision q-ABDHE challenge $(g', g'_{n+2}, g_1, ..., g_n, Z)$, where Z is either $e(g_{q+1}, g')$ or a random element of $\mathbb{G}_T$. $\mathcal{B}$ generates a random polynomial $f(x) \in \mathcal{Z}_p[x]$ of degree q. Suppose $\beta = k\alpha$. It sets $h_1 = g^{f(\alpha)}, h_2 = g^{f(\beta)}$, computing h_1, h_2 from $(g, g_1, ..., g_n)$. $\mathcal{B}$ randomly chooses three hash functions $H_1, H_2, H_3 : \{0,1\}^* \rightarrow \{0,1\}^{logp}$, and then give the public parameters $(g, g_1 = g^\alpha, g_2 = g^{k\alpha}, h_1, h_2)$ to $\mathcal{A}$. Since g, α, and $f(x)$ are chosen uniformly at random, h_1, h_2 are uniformly random and the public parameters have a distribution identical to that in the actual construction.
- **Phase 1:** On $\mathcal{A}'s$ secret key query for set $S_j = \{x_1, x_2, ..., x_n\}$, $\mathcal{B}$ responds to the query as follows. Sets $ID = \sum_{i=1}^{n} H_1(x_i)$. If $ID = \alpha$, $\mathcal{B}$ uses α to solve the truncated decision q-ABDHE problem immediately. Else, let $F_{ID}(x)$ denote a $(q-1)$-degree polynomial $(f(x) - f(ID))/(x - ID)$. $\mathcal{B}$ sets the private key $(r_{ID}, h_{ID_1}), h_{ID_2})$ to be $(f(ID), g^{F_{ID}(\alpha)}, g^{F_{ID}(\beta)})$. This is a valid private key for attribute set $S_j = \{x_1, x_2, ..., x_n\}$, since $ID = \sum_{i=1}^{n} H(x_i)$, $g^{F_{ID}(\alpha)} = g^{(f(\alpha) - f(ID))/(\alpha - ID)} = (h_1 g^{-f(ID)})^{1/(\alpha - ID)}$, and $g^{F_{ID}(\beta)} = g^{(f(\beta) - f(ID))/(\beta - ID)} = (h_2 g^{-f(ID)})^{1/(\beta - ID)}$ as required. Finally, $\mathcal{B}$ sends the secret key $(f(ID), g^{F_{ID}(\alpha)}, g^{F_{ID}(\beta)})$ to $\mathcal{A}$.
- **Challenge:** Eventually adversary $\mathcal{A}$ produces a pair of keyword w_0 and w_1 and a pair of message M_0 and M_1 that it wishes to be challenged on. In addition the adversary $\mathcal{A}$ gives a challenge access structure $\mathbb{A}^* = (w_1, w_2, \cdots, w_n)$. Sets $ID_{\mathbb{A}} = \sum_{i=1}^{n} H_1(w_i)$. If $ID_{\mathbb{A}} = \alpha$, $\mathcal{B}$ uses α to solve the truncated decision q-ABDHE problem immediately. Else, challenger $\mathcal{B}$ computes private key $(f(ID_{\mathbb{A}}), g^{F_{ID_{\mathbb{A}}}(\alpha)}, g^{F_{ID_{\mathbb{A}}}(\beta)})$ for W as in **Phase 1**. Let $f_2(x) = x^{q+2}$ and let $F_{2,ID_{\mathbb{A}}}(x) = (f_2(x) - f_2(ID_{\mathbb{A}}))/(x - ID_{\mathbb{A}})$, which is a polynomial of degree $q + 1$. $\mathcal{B}$ generates bits $b, c \in \{0, 1\}$ and sets the challenging ciphertext as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2916569, IEEE Transactions on Dependable and Secure Computing

follows:

$$
\begin{aligned}
&C_1^* = g'^{f_2(\alpha)-f_2(ID_{\mathbb{A}})}\\
&C_2^* = Z \cdot e(g', \Pi_{i=0}^{q} g^{F_{2,ID_{\mathbb{A}},i}\alpha^i})\\
&C_3^* = M_b/e(C_1, h_{ID_{\mathbb{A}}}) \cdot C_2^{r_{ID_{\mathbb{A}}}}\\
&v^* = H_3(C_1, C_2, C_3)\\
&C_4^* = g_2^v \cdot g^{-v\cdot\Sigma_{i=1}^{n} H_1(w_i)}\\
&C_5^* = e(g,g)^{v^*}\\
&C_6^* = g^{v\cdot H_2(w_c)}
\end{aligned}
$$

where $F_{2,ID_{\mathbb{A}},i}$ is the coefficient of x^i in $F_{2,ID_{\mathbb{A}}}(x)$.

- **Phase 2:** $\mathcal{A}$ repeats the secret key query with the restriction that $\mathcal{A}$ did not previously ask for attribute set S_i which satisfy the challenge access structure W.
- **Guess:** Eventually, $\mathcal{A}$ outputs $b', c' \in \{0, 1\}$.

We note that the above simulations are valid and the keyword of the oracles are uniformly distributed in the keyword space. Hence, the adversary cannot find inconsistence between the simulation and the real world.

Let $s = (log_g g')F_{2,ID_{\mathbb{A}}}(\alpha)$. If $Z = e(g_{q+1}, g')$, then $C_1 = g^{s\cdot(\alpha - ID_{\mathbb{A}})}$, $C_2 = e(g,g)^s$, and $C_3 = M_b/(e(C_1, h_{ID_{\mathbb{A}}}) \cdot C_2^{r_{ID_{\mathbb{A}}}}) = M_b/e(g, h_1)^s$; thus (C_1^*, C_2^*, C_3^*) is valid under randomness s. Meanwhile, v^* is generated from (C_1^*, C_2^*, C_3^*), thus (C_4^*, C_5^*, C_6^*) is also valid under randomness v^*.

Time-complexity: In the simulation, $\mathcal{B}'s$ overhead is dominated by computing $g^{F_{ID}(\alpha)}$ and $g^{F_{ID}(\beta)}$, where $F_{ID}(x)$ is a polynomial of degree $q-1$. Each such computation requires $\mathcal{O}(q)$ exponentiations in $\mathbb{G}$. Let τ_{exp} denote the time complexity to compute one exponentiation, since $\mathcal{A}$ makes at most $q-1$ queries, the time complexity of $\mathcal{B}$ is $\tau' = \tau + o(q^2\tau_{exp})$.

Success probability: If $Z = e(g_{q+1}, g')$, then the simulation is perfect, and $\mathcal{A}$ will guess the bits (b, c) correctly with probability $1/4 + \epsilon'$. Else, Z is uniformly random, and thus (C_2, C_3) is uniformly random and independent element of $\mathbb{G}\times\mathbb{G}_T$. Assuming that no queried $\alpha = \sum_{i=1}^{n} H_1(w_i)$, we see that $|\ pr[\mathcal{B}(g', g'_{n+2}, g_1, ..., g_q, Z) = 0 - 1/4]\ | \leq 2/p$. However, we have that $|\ pr[\mathcal{B}(g', g'_{n+2}, g_1, ..., g_q, Z) = 0 - 1/4]\ | \geq \epsilon'$. Thus, for uniformly random g, g', α and Z, we have that

$$
\begin{aligned}
&|\ pr[\mathcal{B}(g', g'_{q+2}, g_1, ..., g_q, e(g_{q+1}, g')) = 0]\\
&\quad - pr[\mathcal{B}(g', g'_{q+2}, g_1, ..., g_q, Z) = 0]\ | \geq \epsilon' - 2/p.
\end{aligned}
$$

□

As shown in the above theorem, we can conclude that our proposed scheme is semantically secure. Note that our scheme can prevent the cloud server or malicious users from learning any useful information about the encrypted documents, indexes and the trapdoors.

5 EXPERIMENTAL EVALUATION

To evaluate the performance of our scheme, we have measured the efficiency of all the algorithms. Besides, we compare our scheme with other attribute based searchable encryption schemes [36], [40], [44]. We denote system in [36] by HP-ABKS, [40] by ABKS-US, [44] by MVI-ABKS, and our system by FKS-HPABE. We select SHA-3 as the hash function and conduct experiments on a Linux system with an Intel Core 2 Duo CPU at 2.53 GHz-processor and 4GB memory. We exploit the Type A curves within the Paring Based Cryptography (PBC) library [30]. The experimental prototype is written in C language. Type A parings are constructed on the curve $y^2 = x^3 + x$ over the field $\mathcal{Z}_P$ for some prime $p = 3(mod\ 4)$. We suppose $|\mathbb{G}| = |\mathbb{G}_T| = 160$ bits, and $|\mathcal{Z}_P| = 1024$ bits. The experimental results show a single pairing and exponentiation operation in $\mathbb{G}$ cost 28.75 ms and 19.25 ms respectively, and the exponentiation and multiplicative operation in $\mathcal{Z}_P$ cost 0.78 ms and 0.006 ms.

5.1 Functionality Comparisons

We list the key features of our scheme in Table 1 and Table 2 and make a comparison with several related works in e-healthcare cloud computing system in terms of constant overhead, hidden policy and fast keyword search. In order to achieve fine-grained access control, all the schemes adopt ABE technique. From the comparison, we can see that only [36] and [37] support access policy based on multi-value attributes. Schemes [7] [37] [42] and our scheme support constant overhead by deploying "aggregation" technique. In particular, our scheme features a significantly shorter ciphertext size and secret size than that of [36] and [40]. Compared with [7], [37], [42], our scheme obtains the property of constant-size ciphertext. Most importantly, compared with [7], [40] and [42], our scheme achieves policy hiding, which makes our scheme more practical and secure for e-healthcare cloud computing system.

From the comparison, we can see that only [36] and [40] and our scheme support keyword search operation. Unfortunately, as the database grows larger, most schemes do not support efficient keyword search, this will hinder efficient query processing. However, with the "aggregation" technique, the size of ciphertext in our scheme is independent of the number of attributes and the number of attribute values. So, considering the low computing power of resource-constrained mobile devices and healthcare sensors, our scheme with constant overhead is suitable for the e-healthcare cloud computing system.

5.2 Computational Overhead

Let $|E|$ denote a power operation in $\mathbb{G}$, $|P|$ denote a pairing operation in $\mathbb{G}$, $|E_Z|$ denote a power operation in $\mathcal{Z}_P$, $|M_Z|$ denote a multiplication operation in $\mathcal{Z}_P$, n denote the number of attributes in the access policy. In Fig. 4, we show the comparison of execution time of algorithm **Setup** with different values of the variable n. We notice that the computational cost of scheme [36] is $(n+2)|E| + 1|P|$, and that of scheme [40] is $(3n+1)|E| + 1|P|$. While the computational cost of scheme [44] is $3|E| + 1|P|$, and $2|E|$ in our scheme. It is observed that the execution time of algorithms **Setup** in schemes [36] and [40] is linear with the number of attributes in the system. However, the execution time in scheme [44] and in our scheme is constant which is independent with the number of attributes. As shown in Fig. 5, we evaluate the execution time of algorithm **KeyGen** in four schemes by changing the number of attributes from 10 to 100. We notice that the computational cost of scheme [36] is $(2n+2)|E| + n|M_Z|$, in scheme [40] is $(2n+2)|E| + (2n+2)|M_Z|$ and in scheme [44] is $(2n+3)|E| + (2n+1)|M_Z|$. While the computational cost of our scheme is $2|E| + n|E_Z|$. Obviously, the cost of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2916569, IEEE Transactions on Dependable and Secure Computing

TABLE 2: Features comparison with other related works [1]

	[7]	[36]	[37]	[40]	[42]	[44]	Our work
Multi-Value Attributes	×	√	√	×	×	√	√
Constant Overhead	√	×	√	×	√	×	√
Hidden Policy	×	√	√	×	×	×	√
Keyword Search	×	√	×	√	×	√	√
Fast Keyword Search	×	×	×	×	×	√	√
Access Policy [2]	$+,-$	m	m	$+,-,*$	$+,-,*$	m	m

[1] Here "√" and "×" denote either the scheme possesses or does not possess the corresponding property, resp.
[2] "$+,-$" stands for AND-gate on positive and negative attributes, "$+,-,*$" stands for AND-gate on positive and negative attributes with wildcards, "m" stands for AND-gate on multi-value attributes.

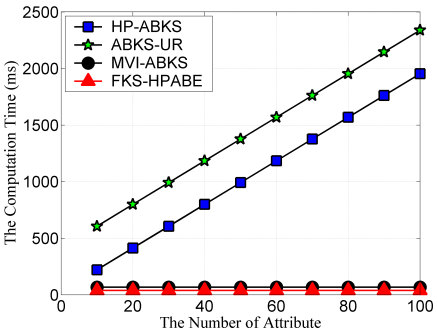

Fig. 4: Performance of Setup

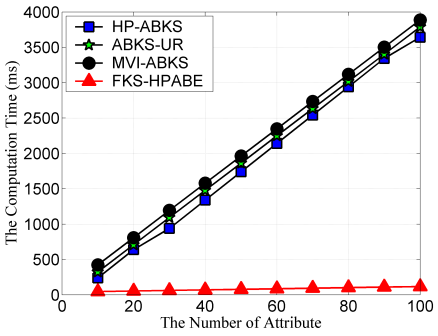

Fig. 5: Performance of KeyGen

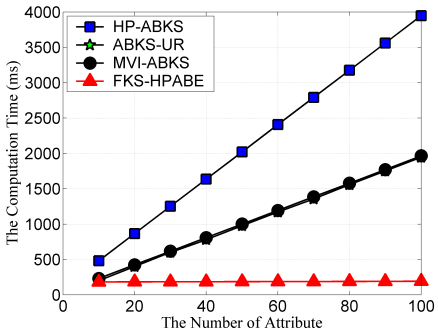

Fig. 6: Performance of Encrypt

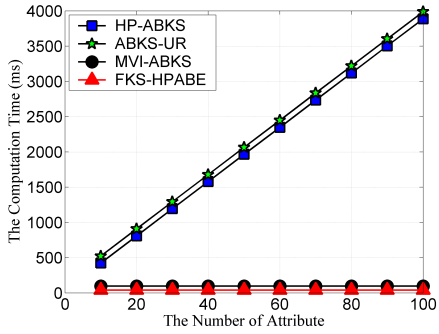

Fig. 7: Performance of Trapdoor

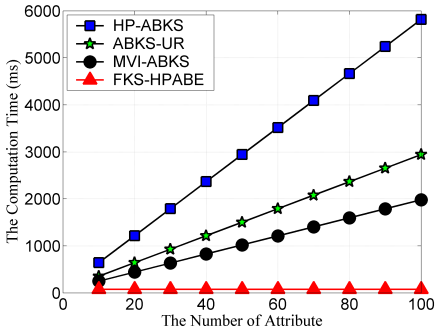

Fig. 8: Performance of Search

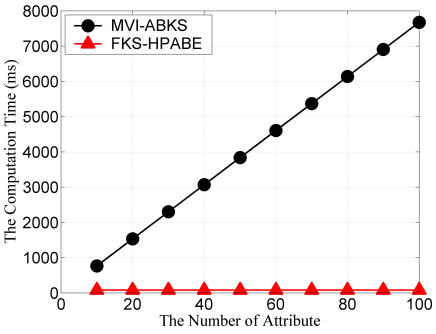

Fig. 9: Performance of Decrypt

generating secret keys in other schemes is linear with the number of attributes in the system. However, the execution time in our scheme is almost constant. In Fig. 6, we show the comparison of execution time of algorithm **Encrypt** with different values of the variable n. We notice that the computational cost of scheme [36] is $(2n+3)|E|$, and that of scheme [40] is $(n+2)|E|+1|M_Z|$. While the computational cost of scheme [44] is $(n+2)|E|+(n+1)|M_Z|$, and $3|P|+8|E|+2n|M_Z|$ in our scheme. It is observed that the execution time of algorithms **Encrypt** in schemes [36] and [40] is linear with the number of attributes in the system. However, the execution time in scheme [44] and in our scheme is almost constant. As shown in Fig. 7, we evaluate the execution time of algorithm **Trapdoor** in four schemes by changing the number of records from 10 to 100. We notice that the computational cost of scheme [36] is $(2n+1)|E|$, and that of scheme [40] is $(2n+2)|E|+2n|M_Z|$. While the computational cost of scheme [44] is $2|E|+2|P|$, and $2|E|+3|M_Z|$ in our scheme. It is observed that the execution time of algorithms **Trapdoor** in schemes [36] and [40] is linear with the number of attributes in the system. However, the execution time in scheme [44] and in our scheme is constant which is independent with the number of attributes. Similarly, we demonstrate the computational costs of the algorithm **Search** by varying the number of records from 10 to 100. Scheme [36] requires $(2n+2)|P|+1|E|$, and scheme [40] needs $(n+1)|P|+2|E|$ whereas that of scheme [44] needs $3|P|+n|M_Z|$. However, our scheme only needs $1|E|+2|P|$. The results are shown in Fig.8, it is observed that the execution time of our scheme is constant, while that of other schemes is linear with the number of attributes in the system. Since in existing schemes [36] and [40], they don't have algorithm **Decrypt**, so we only compare the result between scheme [44] and our proposed scheme. Scheme [44] needs $(2n+1)|P|+n|M_Z|$ whereas that of our scheme needs $1|P|+1|E|+2|M_Z|$. The results are shown in Fig.9, it is observed that the execution time of our scheme is constant, while that of scheme [44] is linear with the number of attributes in the system.

According to the results presented above, we verified that our proposed scheme is more efficient and feasible in practice.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2916569, IEEE Transactions on Dependable and Secure Computing

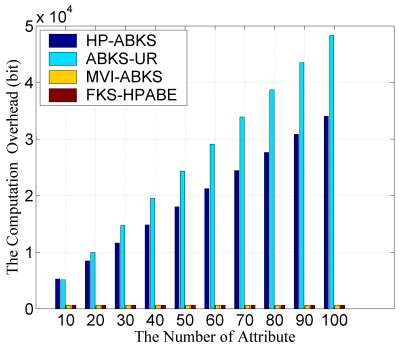

Fig. 10: Computation overhead of Setup

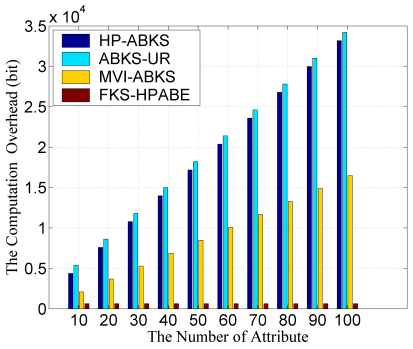

Fig. 11: Computation overhead of KeyGen

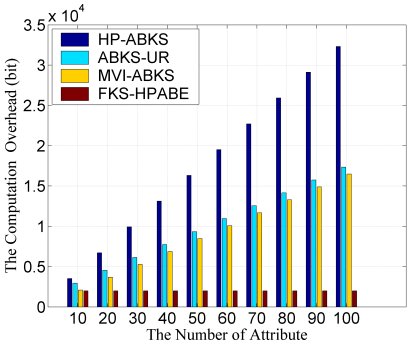

Fig. 12: Computation overhead of Encrypt

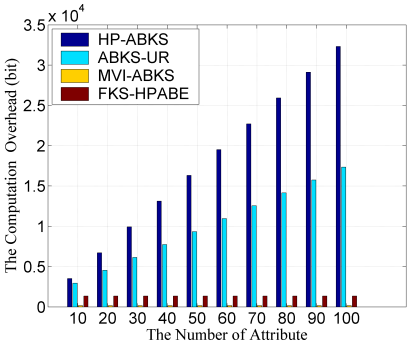

Fig. 13: Computation overhead of Trapdoor

5.3 Storage Overhead

Let $|\mathbb{G}|$ denote a bit length of an element of $\mathbb{G}$ and $|\mathcal{Z}_P|$ denote a bit length of an element of $\mathcal{Z}_P$.

We mainly focus on the storage overhead loaded on user's side. The storage overhead of related schemes are summarized in Fig. 10, Fig. 11, Fig. 12 and Fig. 13 in terms of algorithms **Setup**, **KeyGen**, **Encrypt** and **Trapdoor** (we denote the schemes in [36] by HP-ABKS, [40] by ABKS-US, [44] by MVI-ABKS, and our work by FKS-HPABE). For algorithm **Setup**, schemes [36] and [40] need to store $2n|\mathbb{G}|+2|\mathcal{Z}_P|$ and $(3n+2)|\mathbb{G}|$ respectively. However, scheme [44] and our scheme only need to keep $4|\mathbb{G}|$. $(2n+1)|\mathbb{G}|+1|\mathcal{Z}_P|$ needs to be stored for algorithm **KeyGen** for scheme [36], $(2n+1)|\mathbb{G}|+2|\mathcal{Z}_P|$ for scheme [40] and $(n+3)|\mathbb{G}|$ for scheme [44]. However, in our scheme only $2|\mathbb{G}|$ needs to be stored. In scheme [36], the cloud server needs to store $(2n+2)|\mathbb{G}|$ for algorithm **Encrypt**, $(n+2)|\mathbb{G}|+1|\mathcal{Z}_P|$ in scheme [40] and $(n+3)|\mathbb{G}|$ for scheme [44]. Only $6|\mathbb{G}|+1|\mathcal{Z}_P|$ needs to be stored in our scheme. Similarly, for algorithm **Trapdoor**, user in the system has to store $(2n+2)|\mathbb{G}|$ and $(n+2)|\mathbb{G}|+1|\mathcal{Z}_P|$ in schemes [36] and [40] respectively. While, in scheme [44], only $1|\mathbb{G}|$ needs to be stored. Compare to scheme [44], the storage cost is also constant with little sacrifice which is $2|\mathbb{G}|+1|\mathcal{Z}_P|$.

5.4 Discussion

As shown in Fig. 2, in the searchable e-healthcare system with fine-grained access control, many lightweight devices frequently communicate and transmit data in real-time. So, the designed scheme must also be lightweight and the underlying operations must be computationally efficient. In order to achieve fine-grained access control, we adopt ABE technique as with many existing schemes. However, as was discussed previously, most of the existing ABSE schemes incur large computation and storage costs. Specifically, the size of ciphertext and trapdoor are linear with the number of attributes involved in the access policy. However, with the aggregation technology, our scheme only requires constant computational overhead and constant storage overhead. Thus our proposed scheme is more suitable for the resource limited e-healthcare cloud computing system.

6 EXTENSION

In theory, searchable encryption schemes without information leakage can be designed by using Oblivious RAM, homomorphic encryption and other technologies. However, these methods usually have high complexity and poor practicability. Therefore, in order to design schemes with strong applicability, it is usually necessary to leak certain information. What information is leaked? How much information is leaked? These problems have always been the difficulty of researching searchable encryption technology. Typically, the leaked information includes search pattern (whether two search trapdoors are about the same keyword) and access pattern (the search results). Unfortunately, the disclosure of such information is often fatal.

Recently, researchers have tried to obtain relevant sensitive information from these leaked information. Islam *et al.* [19] and Xu *et al.* [48] recovered user's query keyword with the leaked search pattern and access pattern. Bost *et al.* [4] and Cash *et al.* [6] proved that an attacker can recover all keywords with only a small amount of information leaked from files. Liu *et al.* [27], Yao *et al.* [50] and Ning *et al.* [35] proposed attack methods for keyword domain

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2916569, IEEE Transactions on Dependable and Secure Computing

recovery according to keyword distribution. What's more, Zhang *et al.* [53] introduced an active attack, file injection attack, which allows the attacker to actively and intentionally select specific keywords to inject into the encrypted database, thus effectively recover the query keywords. Generally, an attacker can use his priori knowledge of all or partial of the database to launch an attack. With these priori knowledge, the attacker can infer the distribution information of keywords, thus recovering keywords and database data. Subsequently, researchers put forward corresponding countermeasures such as padding, semantic filtering, batching update and confusing keywords distribution. However, most of the above schemes ignore the privacy protection of search trapdoor in SE system.

ABSE achieving similar properties to public key encryption with keyword search (PEKS) while extending it to the attribute based setting. In most PEKS schemes, secret key is a random number $\alpha \in \mathbb{Z}_p^*$, and the corresponding public key is g^α. Given secret key α and a keyword w, the search user generates a trapdoor as $t_w = H(w)^\alpha$. Based on the Discrete Logarithm Problem, given t_w, it is hard to compute the secret key α. So, given a trapdoor t_w of keyword w, it is hard to compute trapdoor $t_{w'}$ of keyword w'.

In an ABE scheme, we present g^α to simulate the secret key which is mainly composed of group elements, and the corresponding public key is $e(g,g)^\alpha$. Ciphertext can be formed as $(e(g,g)^{\alpha \cdot s}, g^s)$ abstractly, where s is randomly number chosen from $\mathbb{Z}_p^*$. Similarly, in an ABSE scheme, index can be formed as $(e(g,g)^{\alpha \cdot s \cdot H(w)}, g^s)$, and then how to securely generate trapdoor can be a problem:

(1) By generating trapdoor deterministically, a trapdoor can be generated as $g^{\alpha \cdot H(w')}$. Then the cloud server can compute $e(g^{\alpha \cdot H(w')}, g^s) \stackrel{?}{=} (e(g,g)^{\alpha \cdot s \cdot H(w)})$ to search over encrypted data. However, no deterministic encryption scheme can be secure against chosen plaintext attack.

(2) By generating trapdoor probabilistically, given secret key g^a and a keyword w, we can generate a trapdoor as $(g^{\alpha \cdot H(w) \cdot t}, g^t)$, where t is randomly number chosen from $\mathcal{Z}_p^*$. With one group element from $\mathbb{G}_T$ and three group elements from $\mathbb{G}$, it is infeasible to pairing, let alone cloud server to carry out the search operation.

In schemes [9], [24], they use the naive way to generate trapdoor: $t_w = sk \cdot H(w)$. As we know, no deterministic encryption scheme can be secure against chosen-plaintext attack. In other words, any CPA-secure encryption scheme must be probabilistic. Given secret key g^α and a keyword w will raise a question: How to securely generate probabilistic trapdoor? The trapdoor algorithm can be regarded as an encryption function of keyword w and secret key sk. Hence, both of the index and trapdoor should be generated by probabilistic algorithms. However, even with the probabilistic algorithm, the ABSE scheme still can be insecure. We first introduce *trapdoor malleability attack* where the cloud server is given a trapdoor of keyword w by the search user, and the server's goal is to deduce a valid trapdoor of a different keyword w' as shown in Fig. 14.

We now consider the following steps:

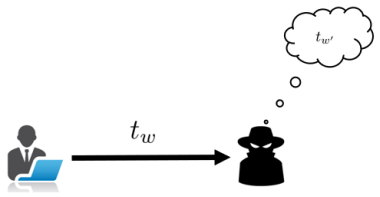

Fig. 14: Trapdoor Malleability Attack

(1) Given a valid trapdoor $t_w = (g^{\alpha \cdot H(w) \cdot t}, g^t)$ of keyword w, the cloud server try to compute searchable ciphertext $C_{w_i} = ABSE(PK, w_i)$ for each keyword. Because keywords are limited, cloud server can get all keyword searchable ciphertext. Since the trapdoor has verification functionality, cloud server can verify whether the keyword in ciphertext is the same as keyword in search trapdoor and the probability of cloud server outputs $w = w_i$ is overwhelming.

(2) Once the cloud server get keyword w, he can compute the secret search key $g^{\alpha \cdot t}$ by computing the $H(w)$-root of $g^{\alpha \cdot H(w) \cdot t}$. And then he can compute trapdoor of any keyword by computing $t'_w = (g^{\alpha \cdot H(w') \cdot t}, g^t)$.

We can see that $t'_w = (g^{\alpha \cdot H(w') \cdot t}, g^t)$ is a valid trapdoor for keyword w' with the random number t. Hence, with the *trapdoor malleability attack* an attacker can forge a valid search trapdoor.

Theorem 2. *Our scheme can resist trapdoor malleability attack: the malicious user is unable to deduce a valid trapdoor of keyword w' when given a valid trapdoor of keyword w.*

Proof. A malicious user $\mathcal{A}$ who owns a trapdoor $t_w = (td_1 = h_{ID_2}^{t \cdot H_2(w)}, td_2 = r_{ID} \cdot t \cdot H_2(w), td_3 = h_2^t)$ of keyword w with secret key $sk = (r_{ID}, h_{ID_1}, h_{ID_2})$, where $h_{ID_1} = (h_1 g^{-r_{ID}})^{\frac{1}{\alpha - \Sigma_{i=1}^n H_1(x_i)}}$ and $h_{ID_2} = (h_2 g^{-r_{ID}})^{\frac{1}{\beta - \Sigma_{i=1}^n H_1(x_i)}}$. The malicious user may try to generate a new trapdoor for w' with the same secret key $sk = (r_{ID}, h_{ID_1}, h_{ID_2})$. To generate a new trapdoor, the malicious should know the value of $(r_{ID}, h_{ID_1}, h_{ID_2})$, which are blinded with the random number t. Based on discrete logarithm problem, each components of secret key is protected by the random number t, so that $\mathcal{A}$ is unable to achieve his goal. □

7 CONCLUSIONS

We presented a hidden policy attribute based encryption scheme in the e-healthcare cloud computing system with constant overhead, which supports fast keyword search. With our new system, the personal health records owner (patient) realizes a fine-grained authorization of data user (doctor) by specifying an access policy in the ciphertext. The cloud server can operate search over encrypted PHRs on behalf of the data user without learning information about the keyword or access policy. Specifically, we achieved the property of fast keyword search in the system, which could narrow down search time in the massive data storage system. We also obtained the property of constant storage

overhead in the system, where the ciphertext size and secret key size do not grow linearly with the size of attribute. We demonstrated security of our scheme through a rigorous proof, and the performance analysis confirmed that our proposed scheme is efficient and practical.

REFERENCES

[1] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE symposium on security and privacy (SP'07)*, IEEE, 2007, pp. 321–334.

[2] I. F. Blake, G. Seroussi, and N. Smart. Advances in elliptic curve cryptography, volume 317 of london mathematical society lecture note series. *Cambridge University Press, Cambridge*, 19(20), 2005, pp. 666.

[3] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2004, pp. 506–522.

[4] R. Bost and P.-A. Fouque. Thwarting leakage abuse attacks against searchable encryption-a formal approach and applications to database padding. *IACR Cryptology ePrint Archive*, 2017, 2017, pp. 1060.

[5] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Transactions on parallel and distributed systems*, 25(1), 2014, pp. 222–233.

[6] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart. Leakage-abuse attacks against searchable encryption. In *ACM Sigsac Conference on Computer and Communications Security*, 2015, pp. 668–679.

[7] C. Chen, Z. Zhang, and D. Feng. Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost. In *International Conference on Provable Security*, Springer, 2011, pp. 84–101.

[8] C. Chen, X. Zhu, P. Shen, J. Hu, S. Guo, Z. Tari, and A. Y. Zomaya. An efficient privacy-preserving ranked keyword search method. *IEEE Transactions on Parallel and Distributed Systems*, 27(4), 2016, pp. 951–963.

[9] B. Cui, Z. Liu, and L. Wang. Key-aggregate searchable encryption (kase) for group data sharing via cloud storage. *IEEE Transactions on Computers*, 65(8), 2016, pp. 2374–2385.

[10] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. In *Proceedings of the 13th ACM conference on Computer and communications security*, ACM, 2006, pp. 79–88.

[11] B. Dan, C. Gentry, S. Halevi, F. Wang, and D. J. Wu. Private database queries using somewhat homomorphic encryption. In *International Conference on Applied Cryptography Network Security*, 2013, pp. 102–118.

[12] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In *International Conference on Information Security Practice and Experience*, Springer, 2009, pp. 13–23.

[13] Z. Fu, F. Huang, K. Ren, W. Jian, and W. Cong. Privacy-preserving smart semantic search based on conceptual graphs over encrypted outsourced data. *IEEE Transactions on Information Forensics Security*, 12(8), 2017, pp. 1874–1884.

[14] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang. Enabling personalized search over encrypted outsourced data with efficiency improvement. *IEEE transactions on parallel and distributed systems*, 27(9), 2016, pp. 2546–2559.

[15] C. Gentry. Practical identity-based encryption without random oracles. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2006, pp. 445–464.

[16] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, ACM, 2006, pp. 89–98.

[17] J. Han, W. Susilo, Y. Mu, and J. Yan. Privacy-preserving decentralized key-policy attribute-based encryption. *IEEE Transactions on Parallel and Distributed Systems*, 23(11), 2012, pp. 2150–2162.

[18] C. Hui, Z. Wan, R. Deng, G. Wang, and Y. Li. Efficient and expressive keyword search over encrypted data in the cloud. *IEEE Transactions on Dependable Secure Computing*, 15(3), 2018, pp. 409–422.

[19] M. S. Islam, M. Kuzu, and M. Kantarcioglu. Access pattern disclosure on searchable encryption: Ramification, attack and mitigation. *Proc Ndss*, 2012, pp. 12.

[20] P. Jiang, Y. Mu, F. Guo, and Q. Wen. Secure-channel free keyword search with authorization in manager-centric databases. *Computers Security*, 69, 2017, pp. 50–64.

[21] Z. Jie, L. Qi, W. Cong, X. Yuan, W. Qian, and K. Ren. Enabling generic, verifiable, and secure data search in cloud services. *IEEE Transactions on Parallel Distributed Systems*, 29(8), 2018, pp. 1721–1735.

[22] M. Kim, H. T. Lee, S. Ling, and H. Wang. On the efficiency of fhe-based private queries. *IEEE Transactions on Dependable Secure Computing*, 15(2), 2018, pp. 357–363.

[23] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2010, pp. 62–91.

[24] J. Li, X. Lin, Y. Zhang, and J. Han. Ksf-oabe: outsourced attribute-based encryption with keyword search function for cloud storage. *IEEE Transactions on Services Computing*, 10(5), 2017, pp. 715–725.

[25] M. Li, S. Yu, N. Cao, and W. Lou. Authorized private keyword search over encrypted data in cloud computing. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, IEEE, 2011, pp. 383–392.

[26] K. Liang, W. Susilo, and J. K. Liu. Privacy-preserving ciphertext multi-sharing control for big data storage. *IEEE Transactions on Information Forensics Security*, 10(8), 2015, pp. 1578–1589.

[27] C. Liu, L. Zhu, M. Wang, and Y. A. Tan. Search pattern leakage in searchable encryption: Attacks and new construction. *Information Sciences An International Journal*, 265(5), 2014, pp. 176–188.

[28] W. Liu, J. Liu, Q. Wu, B. Qin, and Y. Zhou. Practical direct chosen ciphertext secure key-policy attribute-based encryption with public ciphertext test. In *European Symposium on Research in Computer Security*, Springer, 2014, pp. 91–108.

[29] Z. Liu, Z. Cao, and D. S. Wong. Traceable cp-abe: how to trace decryption devices found in the wild. *IEEE Transactions on Information Forensics and Security*, 10(1), 2015, pp. 55–68.

[30] B. Lynn. Pbc library. *Online: http://crypto. stanford. edu/pbc*, 2006.

[31] J. Ning, Z. Cao, X. Dong, K. Liang, H. Ma, and L. Wei. Auditable σ-time outsourced attribute-based encryption for access control in cloud computing. *IEEE Transactions on Information Forensics and Security*, 13(1), 2018, pp. 94–105.

[32] J. Ning, Z. Cao, X. Dong, and L. Wei. White-box traceable CP-ABE for cloud storage service: How to catch people leaking their access credentials effectively. *IEEE Transactions on Dependable and Secure Computing*, 15(5), 2018, pp. 883–897.

[33] J. Ning, Z. Cao, X. Dong, L. Wei, and X. Lin. Large universe ciphertext-policy attribute-based encryption with white-box traceability. In *European Symposium on Research in Computer Security*, Springer, 2014, pp. 55–72.

[34] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin. White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes. *IEEE Transactions on Information Forensics and Security*, 10(6), 2015, pp. 1274–1288.

[35] J. Ning, J. Xu, K. Liang, F. Zhang, and E.-C. Chang. Passive attacks against searchable encryption. *IEEE Transactions on Information Forensics and Security*, 14(3), 2019, pp. 789–802.

[36] S. Qiu, J. Liu, Y. Shi, and R. Zhang. Hidden policy ciphertext-policy attribute-based encryption with keyword search against keyword guessing attack. *Science China Information Sciences*, 60(5), 2017, pp. 052105.

[37] Y. S. Rao and R. Dutta. Recipient anonymous ciphertext-policy attribute based encryption. In *International Conference on Information Systems Security*, Springer, 2013, pp. 329–344.

[38] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2005, pp. 457–473.

[39] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on*, IEEE, 2000, pp. 44–55.

[40] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li. Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, IEEE, 2014, pp. 226–234.

[41] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li. Protecting your right: verifiable attribute-based keyword search with fine-grained

owner-enforced search authorization in the cloud. *IEEE Transactions on Parallel and Distributed Systems*, 27(4), 2016, pp. 1187–1198.
[42] P. V. X. Tran, T. N. Dinh, and A. Miyaji. Efficient ciphertext-policy abe with constant ciphertext length. In *Computing and Convergence Technology (ICCCT), 2012 7th International Conference on*, IEEE, 2012, pp. 543–549.
[43] Z. Wan and R. H. Deng. Vpsearch: Achieving verifiability for privacy-preserving multi-keyword search over encrypted cloud data. *IEEE Transactions on Dependable Secure Computing*, 15(6), 2018, pp. 1083–1095.
[44] H. Wang, X. Dong, and Z. Cao. Multi-value-independent ciphertext-policy attribute based encryption with fast keyword search. *IEEE Transactions on Services Computing*, 2017, pp. 1–11.
[45] H. Wang, X. Dong, Z. Cao, D. Li, and N. Cao. Secure key-aggregation authorized searchable encryption. *Science China Information Sciences*, 62(3), 2019, pp. 39111.
[46] Q. Wang, M. He, M. Du, S. S. M. Chow, and Q. Zou. Searchable encryption over feature-rich data. *IEEE Transactions on Dependable Secure Computing*, 15(3), 2018, pp. 496–510.
[47] Z. Xia, X. Wang, X. Sun, and Q. Wang. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. *IEEE Transactions on Parallel and Distributed Systems*, 27(2), 2016, pp. 340–352.
[48] L. Xu, X. Yuan, C. Wang, Q. Wang, and C. Xu. Hardening database padding for searchable encryption. *IEEE Infocom*, 2019.
[49] R. Xu, K. Morozov, Y. Yang, J. Zhou, and T. Takagi. Efficient outsourcing of secure k-nearest neighbour query over encrypted database. *Computers Security*, 69, 2017, pp. 65–83.
[50] J. Yao, Y. Zheng, C. Wang, and X. Gui. Enabling search over encrypted cloud data with concealed search pattern. *IEEE Access*, 6, 2018, pp. 11112–11122.
[51] X. Yuan, X. Wang, W. Cong, C. Yu, and S. Nutanong. Privacy-preserving similarity joins over encrypted data. *IEEE Transactions on Information Forensics Security*, 12(11), 2017, pp. 2763–2775.
[52] L. Y. Zhang, Y. Zheng, J. Weng, C. Wang, Z. Shan, and K. Ren. You can access but you cannot leak: Defending against illegal content redistribution in encrypted cloud media center. *IEEE Transactions on Dependable Secure Computing*, 2018.
[53] Y. Zhang, J. Katz, U. O. Maryland, and C. Papamanthou. All your queries are belong to us: The power of file-injection attacks on searchable encryption. In *25th USENIX Security Symposium (USENIX Security 16*, 2016, pp. 707–720.
[54] Q. Zheng, S. Xu, and G. Ateniese. Vabks: verifiable attribute-based keyword search over outsourced encrypted data. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, IEEE, 2014, pp. 522–530.
[55] Y. Zheng, H. Cui, W. Cong, and J. Zhou. Privacy-preserving image denoising from external cloud databases. *IEEE Transactions on Information Forensics Security*, 12(6), 2017, pp. 1285–1298.
[56] L. Zhou, Y. Zhu, and A. Castiglione. Efficient k-nn query over encrypted data in cloud with limited key-disclosure and offline data owner. *Computers Security*, 69, 2017, pp. 84–96.

Haijiang Wang received M.S. degrees from Zhengzhou University in 2013, and received Ph.D. degree from Shanghai Jiao Tong University in 2018. He is currently a teacher in the School of Information and Electronic Engineering, Zhejiang University of Science and Technology. His research interests include cryptography and information security, in particular, Public Key Encryption, Attribute-Based Encryption, Searchable Encryption.

Jianting Ning received the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University in 2016. He is currently a research fellow at Department of Computer Science, National University of Singapore. His research interests include applied cryptography and information security, in particular, Public Key Encryption, Attribute-Based Encryption, and Secure Multi-party Computation.

Xinyi Huang received the Ph.D. degree from the School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW, Australia, in 2009. He is currently a Professor with the Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, China. He has authored or co-authored over 60 research papers in refereed international conferences and journals. His research interests include cryptography and information security. He is on the Editorial Board of the International Journal of Information Security (Springer) and has served as the program/general chair or program committee member in over 40 international conferences.

Guiyi Wei received his PhD in December 2006 from Zhejiang University. He has research interests in wireless networks, mobile computing, cloud computing, social networks and network security. He is a full professor of the School of Information and Electronic Engineering at Zhejiang University of Science and Technology. He is also the director of the Networking and Distributed Computing Laboratory.

Geong Sen Poh received his PhD degree in Information Security from Royal Holloway, University of London, UK. He is now a R & D manager at NUS-Singtel Cyber Security Lab. His main research interests include searchable encryption, cryptographic schemes for computations in the encrypted domain, protocols for distributed systems and privacy-preserving data sharing and integration. He was a committee member in the ISO standard for cryptography working group (Malaysia chapter), and committee members for various international conferences.

Ximeng Liu (S'13-M'16) received the B.Sc. degree in electronic engineering from Xidian University, Xi'an, China, in 2010 and Ph.D. degrees in Cryptography from Xidian University, China, in 2015. Now, he is a full professor at College of Mathematics and Computer Science, Fuzhou University, China. Also, he is a research fellow at School of Information System, Singapore Management University, Singapore. He has published over 100 research articles include IEEE TIFS, IEEE TDSC, IEEE TC, IEEE TII IEEE TSC and IEEE TCC. His research interests include cloud security, applied cryptography and big data security.