IEEE Access and Transaction on Cloud Computing,Volume:10,Issue Date:24.March.2022

DSAS: A Secure Data Sharing and Authorized Searchable Framework for e-Healthcare System

LINLIN XUE
School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China
e-mail: 119029@zust.edu.cn

This work was supported by the National Natural Science Foundation of China under Grant 61405178.

ABSTRACT In e-healthcare system, an increasing number of patients enjoy high-quality medical services by sharing encrypted personal healthcare records (PHRs) with doctors or medical research institutions. However, one of the important issues is that the encrypted PHRs prevent effective search of information, resulting in the decrease of data usage. Another issue is that medical treatment process requires the doctor to be online all the time, which may be unaffordable for all doctors (e.g., to be absent under certain circumstances). In this paper, we design a new secure and practical proxy searchable re-encryption scheme, allowing medical service providers to achieve remote PHRs monitoring and research safely and efficiently. Through our scheme DSAS, (1) patients' healthcare records collected by the devices are encrypted before uploading to the cloud server ensuring privacy and confidentiality of PHRs; (2) only authorized doctors or research institutions have access to the PHRs; (3) Alice (doctor-in-charge) is able to delegate medical research and utilization to Bob (doctor-in-agent) or certain research institution through the cloud server, supporting minimizing information exposure to the cloud server. We formalize the security definition and prove the security of our scheme. Finally, performance evaluation shows the efficiency of our scheme.

INDEX TERMS Proxy re-encryption, proxy invisibility, searchable encryption, mobile healthcare sensor networks.

I. INTRODUCTION

Nowadays, with the rapid development of artificial intelligence and the advancement of wearable devices and sensors, e-healthcare sensor network has reached a stage of maturity for adoption and deployment at a commercial scale. E-healthcare sensor network serving as a mobile platform profoundly benefit patients to obtain medical treatment of high quality and efficiency. As shown in Fig.1, patients' devices collect a large amount of personal healthcare records through sensor devices, which enable doctors to more effectively diagnose and attend to the need of the patients through utilizing these data. Such information also enables medical researchers and analysts to perform analytics to gain better insights on illnesses and devise better treatments. Nevertheless, these data may be stored on cloud storage provided by third-party service providers [10], [16], [34], which introduce potential security issues such as data leakage. This is because neither the patients nor the doctors have control of the information once the data is outsourced. This means the privacy and confidentiality of these outsourced data should be protected in such an environment. For instance, some medical institutions collect and store a large amount of PHRs on cloud servers and authorize the usage of these data to the Center for Disease Control and Prevention (CDC). To facilitate disease prevention and control, doctors in CDC are allowed to study these data with data mining technology. However, in the process of collecting case information from medical institutions and the implementation of traditional data mining technology, the CDC may inevitably expose sensitive data of patients. How to store manage and retrieve the PHRs securely and efficiently is a great challenge.

E-healthcare system requires stronger security and privacy guarantees for practices in terms of both data and access to data. In order to prevent information leakage from the stored PHRs, all PHRs stored on the cloud should be encrypted [11], [14], [15], [26], [27], [42]–[44]. Although encryption ensures data confidentiality and can be used to address concerns of data privacy and avoids the attacks from malicious users and cloud servers, it also brings inconvenience of usage. For

The associate editor coordinating the review of this manuscript and approving it for publication was Diana Gratiela Berbecaru.

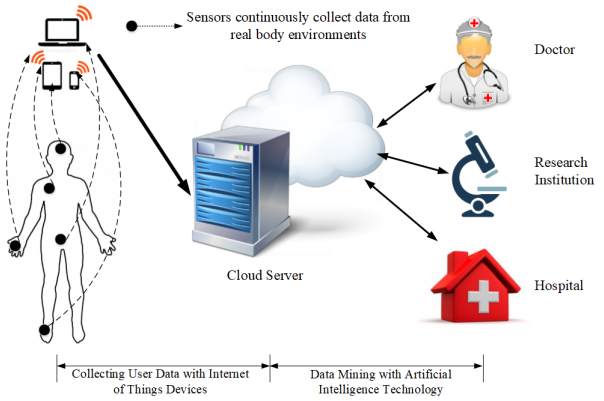

FIGURE 1. Mobile healthcare system.

instance, conventional encryption techniques render it difficult to query these encrypted data [28] because of the useless information retrieval methods based on plaintext. Due to this limitation of conventional, most of the researches employs searchable encryption (SE) cryptosystem to alleviate such concerns. With searchable encryption technology, patients in the e-healthcare system first encrypt the potential keyword as an index and then upload it to the cloud server along with the encrypted PHRs. Then, the authorized doctor or research institution is able to operate encrypted keyword search by sending a trapdoor generated with a certain keyword to the cloud server. With the trapdoor, the cloud server can operate keyword search over the encrypted index and retrieve the corresponding records. Overall, a searchable encryption cryptosystem allows the cloud server to search encrypted data on behalf of users without learning about keywords or plaintext.

With searchable encryption technology, doctors in CDC are able to perform information retrieval over encrypted PHRs and carry out medical treatment. Nevertheless, such a system also implies the doctors need to be available all the time. If the doctor is offline, then medical treatment would not be possible. Proxy re-encryption (PRE) [4], [5], [36] was proposed to solve the above problem by allowing a trusted proxy to securely transform ciphertext belonging to one doctor to another so that a doctor can delegate the medical treatment right to the other doctor in his absent. For instance, suppose there are two doctors Alice and Bob. Each patient with Alice's public key can encrypt the healthcare records to Alice. Suppose Alice is on vacation and wishes to delegate the decryption right to Bob. With PRE technology, Alice generates a re-encryption key based on his private key and Bob's public key, so that with the re-encryption key, the proxy can re-encrypt a ciphertext encrypted under Alice's public key into a ciphertext of the same message under Bob's public key. However, there are two problems with the existing PRE approach. First, the proxy is too powerful: With the re-encryption key, the proxy can transform all ciphertexts of Alice no matter which keyword the ciphertext has. Second, inherent from the bidirectional property, it is impossible to provide collusion-resistance when the dishonest proxy colludes with the delegatee to export the delegator's private key, which constitutes a serious security issue to the system since now the delegatee can impersonate as the delegator. Therefore, it is necessary to restrict the power of proxy server.

A conditional proxy re-encryption searchable (CPRE) [21], [49] system can be deployed to overcome the above issue. In the CPRE system, the delegator generates the re-encryption key with a condition that aims to specify the ciphertext that satisfies the condition. Unfortunately, most existing CPRE schemes cannot guarantee the privacy of the condition, which also contains some sensitive information. On the other hand, if a malicious user can distinguish a re-encrypted ciphertext from an original ciphertext, it will increase the security risk such as that the malicious user knows Alice is not available right now. Thus, it is required for conditional proxy re-encryption to be proxy-invisible, where a malicious user cannot distinguish between the original ciphertext and the re-encrypted ciphertext.

In summary, existing solutions apply many methods (e.g., searchable encryption, proxy re-encryption, conditional proxy re-encryption) sharing PHRs with doctors or medical research institutions to protect data privacy. However, information retrieve over the encrypted PHRs is still a challenging issue, especially when dealing with massive data at a fine-grained level.

A. RELATED WORK

With the rapid development of cloud computing, more and more patients are willing to move their PHRs to the cloud server to enjoy convenient service [25], [30], [35]. To protect data security and personal information privacy, these PHRs are usually stored with encrypted form in the cloud. However, data encryption hinders effective data utilization when the user tries to retrieve files containing some interesting keywords. Yasnoff [48] proposed a e-healthcare storage framework to eliminate the potential for loss of an entire centralized dataset from a single intrusion while maintaining reasonable search performance. A reliable, searchable and privacy-preserving e-healthcare system was proposed by Yang *et al*. [45] based on searchable encryption [9], [18], [23], [40], [51] to protect sensitive healthcare files on cloud storage and enable cloud server to search on the encrypted data under the control of patients. The notion of public-key encryption with keyword search (PEKS) was proposed by Boneh *et al*. [8], who also gave the first PEKS construction for e-healthcare system in the public key environment. Later, Abdalla *et al*. [1] revisited the concept of PEKS and proposed the consistency notion. Baek *et al*. [3] extended PEKS which removes secure channels between a user and the cloud server, which make the patients communicate with doctors with a secure way. More expressive searchable schemes for e-healthcaer system are proposed in [24], [29], [33], and *zhang2017searchable*. To store a huge number of PHRs from multi users, schemes [24], [47] are proposed to optimize data storage and retrieval in the multi user setting.

Except for searchable encryption, proxy re-encryption (PRE) technology proposed by Blaze *et al*. [7] was also employed to store and share medical data in e-healthcare system. Proxy re-encryption is a highly promising solution for cloud computing, which has been widely applied to provide ciphertext transformation in cloud storage services recently. There has been significant progress in PRE over the recent years because of the property called *conditional transformation*, greatly enriching the commercial applications of PRE. In 2005, Ateniese *et al*. [2] proposed a unidirectional scheme and demonstrated how to prevent the proxy from colluding with delegatees in order to expose the delegator's private key. In 2006, Green and Ateniese [17] extended the above notion to identity-based proxy re-encryption, and proposed a new CCA secure scheme. Seo *et al*. [31] proposed the first proxy-invisible CPRE scheme that is secure against CCA secure in the standard model. He *et al*. [19] proposed a non-transferable proxy re-encryption scheme that solves the PKG despotism problem and key escrow problem. Fang *et al*. [12], [13] introduced fuzzy conditional proxy re-encryption and proposed a concrete construction based on the ''set overlap'' distance metric. In [20], PRE was deployed in mobile healthcare social network for a data owner to authorized a healthcare analyzer to access the owner's data. While the underlying purpose is similar, this proposal is more robust using CPRE and examines delegation of duty from a doctor to another, and further provides proxy-invisibility and condition-hiding properties.

Proxy re-encryption with keyword search (PRES), which is proposed by Shao *et al*. [32], can allow the patients to delegate his search and decrypt capability to doctor or research institution. In the e-healthcare system, suppose doctor Alice (delegator) wants to delegate the search capability to doctor Bob (delegatee), by employing the PRES scheme propose by Shao *et al*. [32], 1) Bob can decrypt the ciphertexts delegated from Alice using his own private key; 2) given a trapdoor from Bob, the mail gateway can test whether the ciphertext delegated from Alice contains some special keyword. However, we notice that with the re-encryption key, the proxy can transform all ciphertext of Alice no matter which keyword the ciphertext have. In this case, without Alice's delegation, Bob can still read all the message of Alice, this can be make serious security risks to the e-healthcare system. To address this issue, Weng *et al*. [38], [39] introduced the concept of conditional proxy re-encryption, where the re-encryption key is linked with a condition so that the delegatee can only decrypt ciphertext which satisfying the special condition. After that, a series of CPRE schemes have been proposed [12], [37], [41]. In most CPRE schemes, the condition is specified in the re-encryption key, and thus that the proxy can obtain the condition information such as ''HIV''. However, in the e-healthcare system, the condition can also contain some sensitive information [46]. Therefore, it is necessary to build a CPRE construction without leaking the condition information.

TABLE 1. Functionality summary.

	F1	F2	F3	F4	F5
[4]	√	×	×	×	×
[5]	√	×	×	×	×
[12]	√	×	√	√	√
[21]	√	×	×	×	×
[31]	√	√	×	√	×
[32]	√	×	×	×	√
[36]	√	×	×	×	×
[37]	√	×	×	√	√
[46]	√	×	×	×	√
[49]	√	√	×	×	×

Note: F1: Uni-Directional. F2: Proxy-Invisible. F3: Condition-Hiding. F4: Collusion-Resistance. F5: Keyword Search.

Unfortunately, all the above systems do not simultaneously support both encrypted keyword search and condition-hiding in practice, which limits the commercial applications of proxy re-encryption in the e-healthcare system. We propose a proxy-invisible condition-hiding proxy re-encryption scheme with keyword search to address the issues of inefficiency and condition privacy in the e-healthcare system.

B. MOTIVATION AND CONTRIBUTION

Table 1 gives the summary of the related works in terms of uni-directional, proxy-invisible, condition-hiding, collusion-resistance, keyword search.

- *Uni-Directional*: Uni-directional proxy re-encryption is more superior than multi-directional proxy re-encryption, otherwise, the delegatee may pass permissions to a third party, which will increase the disclosure of privacy. Hence, unidirectionality is a very important characteristic for e-healthcare system.
- *Proxy-Invisible*: In the secure e-healthcare system, if a malicious user can distinguish a re-encrypted ciphertext from an original ciphertext, it will increase the security risk such as the malicious user knows the delegator is not available right now. Hence, e-healthcare system must provide proxy-invisible.
- *Condition-Hiding*: In the conditional proxy re-encryption scheme, the condition often contains some private information. If the condition is exposed, it will cause a great loss to the system. Obviously, if the proxy condition is hidden, the proxy server will get less sensitive information, which makes the e-healthcare system more secure.
- *Collusion-Resistance*: Inherent from trustworthy property, it is impossible to provide collusion-resistance when the dishonest proxy colludes with the delegatee to export the delegator's private key, which would be a disaster to the e-heathcare system. As these authorized work are usually operated on the proxy server (assumed to be a third-party service provider), which for security reason is assumed to be untrusted. Hence, it is necessary to provide collusion-resistance in a secure e-healthcare system.

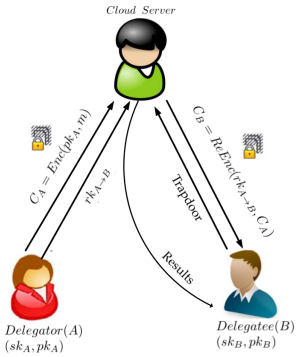

FIGURE 2. System model.

- *Keyword Search:* Encrypting is considered to be a simple and efficient solution to guarantee data confidentiality, but it also makes search over encrypted data extremely difficult. Searchable encryption technology realizes the search operation of encrypted data without decryption, and solves the problem that users cannot control remotely because of data encryption. Hence, searchable is necessary in the e-healthcare system.

As can be seen from Table 1 by giving a comparison among existing shcemes, no scheme can realize secure and reliable ciphertext retrieval functions in the e-healthcare system. In a e-healthcare system, the wearable devices continuously collecting medical data from real body environment. The massive sensitive data leads to a great security and efficiency challenge to the current e-healthcare system due to lack of efficient information retrieve mechanism and poor fine-grained access control. In this paper, we aim to design an efficient, searchable and privacy-preserving e-healthcare system. The overall system consists of three main entities as shown in Fig. 2.

With the proposed infrastructure above, we design a secure data sharing and authorized searchable scheme for e-healthcare system. We show it by exhibiting an example as follow:

As shown in Fig. 2, patients continuously collects PHRs with sensors from physical environments and sends these encrypted PHRs (encrypted under A's public key) to his doctor-in-charge A seeking for medical treatment. In some case, doctor A wants to share some but not all these PHRs to doctor B. To achieve access authorization, A generates a re-encryption key based on his private key and the public key of B. Given the re-encryption key, cloud server is able to convert the ciphertext of A (encrypted under A's public key) to that of B (encrypted under B's public key). Obviously, if there are no restrictions, the cloud server can convert any ciphertext of A, which could make privacy disclosure by sharing unnecessary information. In order to prevent privacy disclosure, we generate a conditional re-encryption by embedding a trapdoor (e.g.,pneumonia) in the re-encryption key so that the cloud server can only convert ciphertext under the designated condition. Moreover, the cloud server is responsible for storing the encrypted data and providing keyword search services and also acts as a proxy to perform re-encryption for data users. When a keyword search request with a trapdoor is received from B, the cloud server performs information retrieval over the encrypted PHRs. Finally, B can decrypt ciphertext by using only his private key to obtain specific medical information.

In summary, users (e.g., patients, doctors, research institutions) enjoy an efficient, searchable, privacy protection service in our e-healthcare system. The main results are as follows:

1. Data privacy: patients' data collected by the sensor devices are encrypted before they are uploaded to the cloud storage server. This ensures privacy and confidentiality of data since the cloud server will not be able to learn any information from the encrypted PHRs.
2. Conditional authorization: In the event where the doctor-in-charge (Alice) is unavailable, our scheme enables the delegation of the task to another doctor (Bob) through a cloud server, without the need to decrypt the PHRs thus minimizing information exposure to the cloud server.
3. Condition-hiding: Our scheme not only guarantees patients's PHRs privacy through encrypted data but also preserves the privacy of the condition embedded in the re-encryption key.
4. Proxy invisibility: In our scheme, the authorized doctor (Bob) or a malicious user cannot distinguish which ciphertext is sent to delegatee and which ciphertext is re-encrypted by the cloud delegated by Alice.
5. Collusion resistance: In our scheme, even a dishonest proxy colludes with Bob, Alice's private key can still be secure.

II. PRELIMINARIES

In this section, we present basic assumptions and cryptographic concepts.

A. BILINEAR MAPS

We briefly review describe bilinear maps and bilinear map groups (For more detail, see [6]). Consider the following setting: $\mathbb{G}$ and $\mathbb{G}_T$ are two multiplicative cyclic groups of prime order p; the group action on $\mathbb{G}$ and $\mathbb{G}_T$ can be computed efficiently; g is a generator of $\mathbb{G}$; and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map. The bilinear map e has the following properties:

- Bilinearity: $\forall u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
- Non-degeneracy: $e(g, g) \neq 1$.

We say that $\mathbb{G}$ is a bilinear group if the group operations in $\mathbb{G}$ and the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ can both be computed efficiently. Notice that the map $e(\cdot, \cdot)$ is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

B. HARD ASSUMPTIONS

The modified Decisional Bilinear Diffie Hellman assumption (mDBDH) problem is defined as follow. Let $\mathbb{G}$ be bilinear group of prime order p and g is a generator of $\mathbb{G}$. Given (g, g^a, g^b, g^c) for random $a, b, c \in \mathbb{Z}_p^*$ and $Z \in \mathbb{G}_T$, decide whether $Z = e(g, g)^{ab/c}$. An algorithm $\mathcal{A}$ that outputs $b \in \{0, 1\}$ has advantage ε in solving the mDBDH problem if

$$|\Pr[\mathcal{B}(g, g^a, g^b, g^c, e(g, g)^{ab/c}) = 0] - \Pr[\mathcal{B}(g, g^a, g^b, g^c, Z) = 0]| \geq \varepsilon,$$

where the probability is over the random choice of $a, b, c \in \mathbb{Z}_p^*$, the random choice $Z \in \mathbb{G}_T$, and the random bits consumed by $\mathcal{A}$. We say that the mDBDH assumption holds in $\mathbb{G}$ if no PPT algorithm has advantage at least ϵ in solving the mDBDH problem.

The q-weak Decisional Bilinear Diffie Hellman Inversion assumption (q-DBDHI) problem is defined as follow, let $\mathbb{G}$ be bilinear group of prime order p and g is a generator of $\mathbb{G}$. Given $(g, g^\alpha, \ldots, g^{\alpha^q}, g')$ for random $\alpha \in \mathbb{Z}_p^*$ and $Z \in \mathbb{G}_T$, decide whether $Z = e(g, g')^{1/\alpha}$. An algorithm $\mathcal{A}$ that outputs $b \in \{0, 1\}$ has advantage ε in solving the mDBDH problem if

$$|\Pr[\mathcal{B}(g, g^\alpha, \ldots, g^{\alpha^q}, g', e(g, g')^{1/\alpha}) = 0] - \Pr[\mathcal{B}(g, g^\alpha, \ldots, g^{\alpha^q}, g', Z) = 0]| \geq \varepsilon,$$

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 $\mathcal{A}$. We say that the q-DBDHI assumption holds in $\mathbb{G}$ if no PPT algorithm has advantage at least ϵ in solving the q-DBDHI problem.

III. SYSTEM ARCHITECTURE AND CONSTRUCTION

In this section, we first introduce the algorithms definition and system architecture, and then propose the construction of our conditional proxy re-encryption with keyword search system, which is proxy-invisible, condition-hiding and CCA-secure.

A. DEFINITION

A conditional proxy re-encryption with keyword search system consists of the following polynomial time algorithms:

- $Setup(1^\lambda) \rightarrow param$: Given a security parameter λ, outputs public parameters $param$ to be used by all parties.
- $KeyGen(1^\lambda) \rightarrow (pk, sk)$: Given a security parameter λ, the key generation algorithm outputs a public/private key pair (pk, sk).
- $Enc(pk, m, w) \rightarrow CT$: Given a public key pk, a keyword w, and a message m, the encryption algorithm outputs a ciphertext CT of m corresponding to keyword w.
- $ReKeyGen(sk_i, pk_j, w) \rightarrow rk^w_{i\rightarrow j}$: Given a user i's private key sk_i, a user j's public key pk_j and condition w, the re-encryption key generation algorithm outputs a re-encryption key $rk^w_{i\rightarrow j}$.
- $ReEnc(rk^w_{i\rightarrow j}, CT_i) \rightarrow CT_j$: Given the re-encryption key $rk^w_{i\rightarrow j}$ and a ciphertext CT_i corresponding public key pk_i, the re-encryption algorithm outputs another ciphertext CT_j corresponding public key pk_j or the special character $\perp$ indicating an error.
- $Trapdoor(sk, w) \rightarrow t_w$: Given a user's private key sk and a keyword w, the trapdoor algorithm outputs a trapdoor t_w of keyword w corresponding to the user.
- $Test(CT, t_w) \rightarrow 0$ *or* 1: Given ciphertext CT, and a trapdoor t_w, the test algorithm outputs 1 if a given ciphertext CT contains the keyword w specified by the trapdoor t_w or 0 otherwise.
- $Dec(sk, CT) \rightarrow m$: Given a user's private key sk and a ciphertext CT, the decryption algorithm outputs the corresponding message m or the special character $\perp$ indicating an error.

B. SYSTEM ARCHITECTURE

In this paper, we design a new cloud storage framework for e-healthcare system which provides efficient and privacy-preserving information retrieve service and meet the above requirements. The e-healthcare system generally consists of the following phases:

- *Setup* phase: In this phase, patients' sensors choose a security parameter 1^λ, run algorithms *Setup* and *KeyGen* to generate and store parameters *param*, public key and private key (pk, sk) for all patients in the real world to collect PHRs.
- *Data collection and encryption* phase: The sensors continuously collect PHRs F from physical environments, then extract keyword w from these data, run algorithm *Enc* to generate medical information under doctor Alice's public key pk_{Alice}. Finally, upload all ciphertext CT_{Alice} to the cloud server.
- *Data conversion* phase: Alice is able to delegate search and decrypt operation to Bob through the cloud server with the following steps if Alice is unavailable. First, Alice runs algorithm *ReKeyGen* to generate re-encryption key for the cloud server under Alice's private key sk_{Alice} and Bob's public key pk_{Bob}. Second, given re-encryption key, the cloud server runs algorithm *ReEnc* to convert the corresponding ciphertext. Finally, stores the converted ciphertext CT_{Bob}. To achieve conditional authorization, algorithm *ReKeyGen* requires Alice's private key as part of the input. Therefore, anyone (without Alice's private key) given Bob's public key could not launch conditional authorization.
- *Data retrival* phase: Bob is able to search and decrypt the converted ciphertext with the following steps. First, Bob runs algorithm *Trapdoor* to generate a trapdoor t_w under keyword w and his private key sk_{Bob}. Second, given the trapdoor t_w and ciphertext CT_{Bob}, the cloud server runs algorithm *Test* to find the matching ciphertext. Finally, Bob obtains the intended data by decrypting the matched ciphertext with his private key sk_{Bob} by running algorithm *Dec*.

Fig. 3 shows the flow chart of the above system. The black arrow in the figure represents the process of data collection:

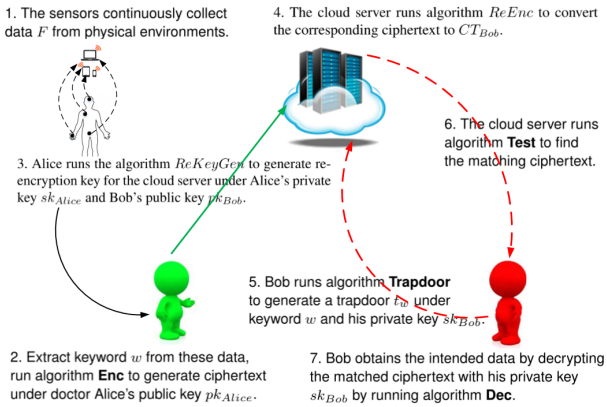

FIGURE 3. Data process, storage, and retrieval in e-healthcare system.

patients' data collected by the sensor devices are encrypted before they are uploaded to the cloud storage server. This ensures privacy and confidentiality of data since the cloud server will not be able to learn any information from the encrypted personal health records (PHRs). The red arrows in figure represent the process of secure data query: Only the authorised doctor have access to the PHRs. In the event where the doctor-in-charge is unavailable, our scheme enables the delegation of task to another doctor through a cloud server, without the need to decrypt the PHRs thus minimising information exposure to the cloud server, which shown with green arrow in the figure.

C. DSAS CONSTRUCTION

Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order p, and let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be the bilinear map. Our conditional proxy re-encryption with keyword search system works as follows.

- $Setup(1^\lambda)$: Given a security parameter λ, the setup algorithm chooses $\mathbb{G}$ and $\mathbb{G}_T$ to be groups of order p with the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ and chooses a strongly unforgeable one-time signature $Sig = (\mathcal{G}, \mathcal{S}, \mathcal{V})$. It picks random generators $g, f, h, u, v \in \mathbb{G}$. Also, two hash functions $H_1 : \{0,1\}^* \to \mathbb{G}$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_p^*$ are selected randomly. The public parameters *param* are given by

$$param = (\mathbb{G}, \mathbb{G}_T, e, g, f, h, u, v, Sig, H_1, H_2)$$

- $KeyGen(1^\lambda)$: On input 1^λ, user i chooses random $x_i, y_i \in \mathbb{Z}_p^*$ and sets $X_i = f^{x_i}$ and $Y_i = h^{y_i}$. The public key and private key of user i are

$$pk_i = (X_i, Y_i), \quad sk_i = (x_i, y_i)$$

- $Enc(pk, m, w)$: To encrypt a message $m \in \mathbb{G}_T$ under the public key pk_i, the data owner selects a one-time signature key pair $(ssk, svk) \leftarrow \mathcal{G}(\lambda)$, picks random $s, r \in \mathbb{Z}_p^*$, and sets

$$\langle C_1 = Y_i^r, C_2 = f^{\frac{1}{r}}, C_3 = f^{\frac{s}{r}}, C_4 = X_i^s \cdot f^{-s \cdot H_2(ID_i)},$$
$$C_5 = e(h, H_1(w))^r, C_6 = (u^{svk} \cdot v)^s,$$
$$C_7 = m \cdot e(f, h)^{-s} \rangle$$

where ID_i is identity of user i. Then, the data owner generates a one-time signature $\sigma = \mathcal{S}(ssk, (C_6, C_7))$, outputs the ciphertext as

$$CT_i = (svk, C_1, C_2, C_3, C_4, C_5, C_6, C_7, \sigma)$$

- $ReKeyGen(sk_i, pk_j, w)$: Given a user i's private key sk_i, a user j's public key pk_j and condition w, user i sets re-encryption key $rk_{i\to j}^w = (rk_1, rk_2, rk_3)$ as

$$rk_1 = Y_j^{\frac{1}{x_i - H_2(ID_i)}}, \quad rk_2 = H_1(w)^{\frac{1}{y_i}},$$
$$rk_3 = e(H_1(w), h)^{\frac{1}{x_i - H_2(ID_i)}}$$

- $ReEnc(rk_{i\to j}^w, CT_i)$: Given the re-encryption key $rk_{i\to j}^w = (rk_1, rk_2, rk_3)$ and a ciphertext $CT_i = (svk, C_1, C_2, C_3, C_4, C_5, C_6, C_7, \sigma)$, the cloud server first checks whether the condition hold by running **Test**. If the outputs is $\perp$ then terminate; Otherwise, the cloud server picks random $t \in \mathbb{Z}_p^*$ and computes

$$\langle C_1' = rk_1^t, C_2' = (X_i \cdot f^{-H_2(ID_i)})^{\frac{1}{t}}, C_3' = C_4^{\frac{1}{t}},$$
$$C_4' = C_4, C_5' = rk_3^t, C_6' = C_6, C_7' = C_7 \rangle$$

The re-encrypted ciphertext for user j is

$$CT_j = (svk, C_1', C_2', C_3', C_4', C_5', C_6', C_7', \sigma)$$

- $Trapdoor(sk, w)$: On input a user i's private key sk_i and a keyword w, output the keyword w's trapdoor as

$$t_w = H_1(w)^{\frac{1}{y_i}}$$

- $Test(CT, t_w)$: Given the trapdoor t_w and ciphertext CT_i, the cloud server checks the the validity of the ciphertext by testing the following relations.
  - $\mathcal{V}(svk, \sigma, (C_6, C_7)) = 1$
  - $e(C_2, C1 \cdot C_6) = e(f, Y_i) \cdot e(u^{svk} \cdot v, C_3)$

  If the check is fails, output $\perp$. Otherwise do the test step: If $e(t_w, C_1) = C_5$, then output "1"; otherwise output "0".
- $Dec(sk, CT)$: On input of a user i's private key sk_i and ciphertext $CT_i = (svk, C_1, C_2, C_3, C_4, C_5, C_6, C_7, \sigma)$, user i first checks the validity of the ciphertext. If the check fails, then output $\perp$. Otherwise, user i output a message by computing

$$m = C_7 \cdot e(C_1, C_3)^{\frac{1}{y_i}}$$

IV. SECURITY DEFINITION AND PROOF

In this section, we give the security definition and the concrete proof of the proposed DSAS scheme.

A. SYSTEM THREAT MODEL

We assume cloud server is always online with sufficient storage and computing capacity. Also, we assume that doctor Alice is online most of the time. In some cases, when Alice is not online, he authorizes access to the PHRs to doctor Bob or other medical institutions by distributing a re-encryption key through a secure channel between cloud server and himself.

However, the possible attacks on our system are as follows:

1. The could server is ''honest-but-curious'', which follows many related work on e-healthcare cloud computing system [12], [32], [35]–[37], which means the cloud server ''honestly'' follows the designated protocol, but ''curiously'' infers additional privacy information of the encrypted PHRs content or the search query.
2. Unlike FSGW [12], SCLL [32], WHYLW [37] and YM [46], the cloud server in our system may collude with authorized doctors to export the delegator's private key to access data beyond their access privileges.

B. SYSTEM SECURITY MODEL

We define security for our system in the sense of semantic-security. We need to ensure that a ciphertext CT does not reveal any information about the keyword w unless the keyword trapdoor t_w is available. We define security against an active attacker who is able to obtain trapdoors t_w for any w of his choice, even under such attack, the attacker should not be able to distinguish encryption of a keyword w_0 from encryption of a keyword w_1 for which he did not obtain the trapdoor. Formally, we define security against an active adversary $\mathcal{A}$ using the following game between a challenger $\mathcal{B}$ and the adversary $\mathcal{A}$.

Game 1: (IND-CKA game: Privacy of keyword)

- **Setup:** Challenger $\mathcal{B}$ runs the *Setup* algorithm and forwards public parameters *param* to adversary $\mathcal{A}$.
- **Phase 1:** Adversary $\mathcal{A}$ can adaptively make the following queries:
  - Uncorrupted key generation oracle $\mathcal{O}_{pk}$: $\mathcal{B}$ obtain a public/private key pair (pk_i, sk_i), and sends pk_i to $\mathcal{A}$.
  - Corrupted key generation oracle $\mathcal{O}_{sk}$: $\mathcal{B}$ obtain a public/private key pair (pk_i, sk_i), and sends (pk_i, sk_i) to $\mathcal{A}$.
  - Trapdoor generation oracle $\mathcal{O}_{td}$: $\mathcal{B}$ runs trapdoor generation to generate a trapdoor t_w and sends it to $\mathcal{A}$.
  - Re-encryption key generation oracle $\mathcal{O}_{rk}$: $\mathcal{B}$ runs re-encryption key generation to generate a re-encryption key $rk_{i\to j}^{w}$ and sends it to $\mathcal{A}$.
  - Re-encryption oracle $\mathcal{O}_{re}$: $\mathcal{B}$ runs re-encryption generation to convert ciphertext CT_i to ciphertext CT_j and sends CT_j to $\mathcal{A}$.
- **Challenge:** At some point, the adversary $\mathcal{A}$ sends the challenger $\mathcal{B}$ two keywords w_0, w_1, a message m and a public key pk^* on which it wishes to be challenged. The challenger $\mathcal{B}$ picks a random $b \in \{0, 1\}$ and gives adversary the challenge index CT_b^*.
- **Phase 2:** Adversary $\mathcal{A}$ can adaptively make more queries as in **Phase 1**:
- **Guess:** Eventually, the adversary $\mathcal{A}$ outputs $b' \in \{0, 1\}$ and wins the game if $b = b'$.

During the above game, adversary $\mathcal{A}$ is subject to the following restrictions where $w^* \in \{w_0, w_1\}$:

1. $\mathcal{A}$ cannot query the target private key sk_{i^*}.
2. $\mathcal{A}$ cannot query the trapdoor of (pk_{i^*}, w^*).
3. $\mathcal{A}$ cannot issue re-encryption key $rk_{i^*\to j}^{w*}$ if pk_j appears in a previous corrupted key generation query.
4. $\mathcal{A}$ cannot issue re-encryption on $pk_{i^*}, pk_j, (w^*, CT^*)$ if pk_j appears in a previous corrupted key generation query.

The advantage of an adversary is defined to be $Adv = |\Pr[b = b'] - 1/2|$ in this game.

Definition 1 (IND-CKA): A conditional proxy re-encryption with keyword search system is semantically secure against an adaptive chosen keyword attack if all probabilistic polynomial-time (PPT) attackers have at most negligible advantage in λ in the above security game.

We now define security for our system to ensure that CT does not reveal any information about M. Formally, we define security against an active adversary $\mathcal{A}$ using the following game between a challenger $\mathcal{B}$ and the adversary $\mathcal{A}$.

Game 2: (IND-CCA game: Privacy of message)

- **Setup:** Challenger $\mathcal{B}$ runs the *Setup* algorithm and forwards public parameters *param* to adversary $\mathcal{A}$.
- **Phase 1:** Adversary $\mathcal{A}$ can adaptively make the following queries:
  1: $\mathcal{A}$ can query $\mathcal{O}_{pk}$, $\mathcal{O}_{sk}$, $\mathcal{O}_{td}$, $\mathcal{O}_{rk}$, $\mathcal{O}_{re}$ identical to that in the security model of Game 1.
  2: **Decryption oracle** $\mathcal{O}_{dec}$: On input pk, CT by $\mathcal{A}$, return the plaintext m.
- **Challenge:** At some point, the adversary $\mathcal{A}$ sends the challenger $\mathcal{B}$ two message m_0, m_1, a conditional keyword w and a public key pk^* on which it wishes to be challenged. The challenger $\mathcal{B}$ picks a random $b \in \{0, 1\}$ and gives adversary the challenge ciphertext CT_b^*.
- **Phase 2:** Adversary $\mathcal{A}$ can adaptively make more queries as in **Phase 1**:
- **Guess:** Eventually, the adversary $\mathcal{A}$ outputs $b' \in \{0, 1\}$ and wins the game if $b = b'$.

During the above game, adversary $\mathcal{A}$ is subjected to the following restrictions where $w^* \in \{w_0, w_1\}$:

1. $\mathcal{A}$ cannot query the target private key sk_{i^*}.
2. $\mathcal{A}$ cannot query the trapdoor of (pk_{i^*}, w^*).
3. $\mathcal{A}$ cannot issue re-encryption key $rk_{i^*\to j}^{w*}$ if pk_j appears in a previous corrupted key generation query.
4. $\mathcal{A}$ cannot issue re-encryption on $pk_{i^*}, pk_j, (w^*, CT^*)$ if pk_j appears in a previous corrupted key generation query.
5. $\mathcal{A}$ cannot issue decryption query on neither pk_{i^*}, CT_i^* nor pk_j, CT_j^*, where pk_j, CT_j^* is a re-encryption of the challenge ciphertext.

The advantage of an adversary is defined to be $Adv = |\Pr[b = b'] - 1/2|$ in this game.

Definition 2 (IND-CCA): A conditional proxy re-encryption with keyword search system is secure against an adaptive chosen ciphertext attack if all probabilistic polynomial-time (PPT) attackers have at most negligible advantage in λ in the above security game.

C. SECURITY PROOF

We now prove that our condition-hiding proxy re-encryption with keyword search system is *IND-CKA* secure based on the *mDBDH* assumption in the random oracle model.

Theorem 1: If the mDBDH assumption holds in $\mathbb{G}$ and $\mathbb{G}_T$, then our construction is IND-CKA *secure in the random oracle model.*

Proof: Suppose there exists a polynomial-time adversary $\mathcal{A}$ that breaks the *IND-CKA* security of system described above with advantage ϵ. We construct an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve *mDBDH* problem with probability $(\frac{1}{2} + \frac{\epsilon}{e \cdot q_{H_1}})$. $\mathcal{B}$ takes as input a random *mDBDH* challenge (g, g^a, g^b, g^c, Z), where Z is either $e(g, g)^{\frac{ab}{c}}$ or random element of $\mathbb{G}_T$. $\mathcal{B}$ interacts with $\mathcal{A}$ in the security game as follows:

- **Setup:** $\mathcal{B}$ selects $u, v \in \mathbb{G}$, a strongly unforgeable one-time signature $Sig = (\mathcal{G}, \mathcal{S}, \mathcal{V})$, $(ssk^*, svk^*) \leftarrow \mathcal{G}(\lambda)$ and $H_1 : \{0,1\}^* \to \mathbb{G}$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_p^*$. Let $h = g^{k_2}, f = (g^b)^{k_1}$, where $k_1, k_2 \in \mathbb{Z}_p^*$. The system public parameters are $(\mathbb{G}, \mathbb{G}_T, e, g, f, h, u, v, Sig, H_1, H_2)$.
  Hash Oracles: $\mathcal{B}$ builds the following oracles. $\mathcal{O}_{H_1}$ On input w to hash function H_1, $\mathcal{B}$ first checks whether tuple $(w, h_1, r_1, coin)$ occurs in table T_{H_1}. If yes, $\mathcal{B}$ responds $\mathcal{A}$ with r_1. Otherwise, $\mathcal{B}$ sets $cion = 0$ with probability $1/(q_{H_1} + 1)$, where $coin \in \{0,1\}$, q_{H_1} is the maximum number of queries of $\mathcal{O}_{H_1}$. Finally, $\mathcal{B}$ responds $\mathcal{A}$ with $h_1 = (g^c)^{r_1}$ if $coin = 1$ or $h_1 = (g^a)^{r_1}$ if $coin = 0$ and records tuple $(w, h_1, r_1, coin)$ in table T_{H_1}. Similarly, for $\mathcal{O}_{H_2}$, $\mathcal{B}$ responds to a query for $H_2(ID)$ by picking a new random value $r_2 \in \mathbb{Z}_p^*$ for each new ID and sets $H_2(ID) = r_2$, and records tuple (ID, r_2) in table T_{H_2}.
- **Phase 1:** On $\mathcal{A}'s$ queries, $\mathcal{B}$ builds the following oracles.
  - $\mathcal{O}_{pk}$: On input an identity ID_i, $\mathcal{B}$ picks random $x_i, y_i \in \mathbb{Z}_p^*$. If $corrupted_i = 1$, $\mathcal{B}$ sets $X_i = (g^b)^{k_1 \cdot x_i}, Y_i = g^{k_2 \cdot y_i}$; otherwise, $\mathcal{B}$ sets $X_i = (g^b)^{k_1 \cdot x_i}$, $Y_i = (g^c)^{k_2 \cdot y_i}$. Finally, $\mathcal{B}$ records the tuple $(X_i, Y_i, x_i, y_i, corrupted_i)$ in table T_k, and responds $\mathcal{A}$ with $pk_i = (X_i, Y_i)$.
  - $\mathcal{O}_{sk}$: On input an identity ID_i, $\mathcal{B}$ checks whether pk_i occurs in T_k, if not, $\mathcal{B}$ terminates. Otherwise, if $corrupted_i = 0$, $\mathcal{B}$ terminates. If $corrupted_i = 1$, $\mathcal{B}$ responds $(pk_i = (X_i, Y_i), sk_i = (x_i, y_i))$.
  - $\mathcal{O}_{td}$: On input (ID_i, w), if $corrupted_i = 1$, $\mathcal{B}$ responds $\mathcal{A}$ with $H_1(w)^{\frac{1}{y_i}}$. If $corrupted_i = 0$, $\mathcal{B}$ get tuple $(w, h_1, r_1, coin)$ in table T_{H_1} and responds $\mathcal{A}$ with $g^{\frac{r_1}{y_i}}$ if $coin = 1$; otherwise, aborts.
  - $\mathcal{O}_{rk}$: On input (ID_i, ID_j, w). $\mathcal{B}$ queries $\mathcal{O}_{td}((ID_i, w))$ to get rk_2. $\mathcal{B}$ get tuple $(X_i, Y_i, x_i, y_i, corrupted_i)$ in table T_k and tuple (ID, r_2) in table T_{H_2}, then responds $\mathcal{A}$ with
  $$rk_1 = Y_j^{\frac{1}{x_i - r_2}}, \quad rk_3 = e(H_1(w), g^{k_2})^{\frac{1}{x_i - r_2}}$$
  - $\mathcal{O}_{re}$: On input identities of the delegator and delegatee and the original ciphertext
  $$\langle ID_i, ID_j, CT_i = (svk, C_1, C_2, C_3, C_4, C_5, C_6, C_7, \sigma)\rangle$$
  $\mathcal{B}$ queries $\mathcal{O}_{rk}(ID_i, ID_j, w)$ to get rk_1 and tuple (ID, r_2) in table T_{H_2}, and picks random $t \in \mathbb{Z}_p^*$ and computes
  $$\langle C_1' = rk_1^t, C_2' = (X_i \cdot f^{-r_2})^{\frac{1}{t}}, C_3' = C_4^{\frac{1}{t}}, C_4' = C_4, C_5' = rk_3^t, C_6' = C_6, C_7' = C_7\rangle$$
  $\mathcal{B}$ responds $\mathcal{A}$ with
  $$CT_j = (svk, C_1', C_2', C_3', C_4', C_5', C_6', C_7', \sigma)$$
- **Challenge:** Eventually adversary $\mathcal{A}$ produces a pair of keywords w_0 and w_1, a message m and a public key pk^* on which it wishes to be challenged. If pk^* is not in table T_k, or $corrupted = 1$, $\mathcal{B}$ terminates. If $coin_0 = 1$ and $coin_1 = 1$, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ chooses a random number $b \in \{0,1\}$ such that $coin_d = 0$ and selects random $s \in \mathbb{Z}_p^*$ and responds $\mathcal{A}$ with the challenge ciphertext CT_b^*:
$$\begin{aligned}\langle C_1^* &= (g^b)^{k_2 \cdot y_i} = (pk^*)^{\frac{b}{c}},\\ C_2^* &= (g^c)^{k_1} = f^{\frac{c}{b}}, C_3^* = (C_2^*)^s,\\ C_4^* &= X_i^s \cdot ((g^b)^{k_1})^{-s \cdot r_2^*},\\ C_5^* &= Z^{r_1 \cdot k_2} = e(g^{k_2}, H_1(w_b))^{\frac{b}{c}}\\ C_6^* &= (u^{svk^*} \cdot v)^s,\\ C_7^* &= m \cdot e(((g^b)^{k_1}), g^{k_2})^{-s}\\ \sigma^* &= \mathcal{S}(ssk^*, (C_6^*, C_7^*))\rangle\end{aligned}$$
  where r_1 is the corresponding value to w_d in table T_{H_1}, r_2 is the corresponding value to ID_{i^*} in table T_{H_2} and y_i is the corresponding value to pk^* in table T_k.
- **Phase 2:** $\mathcal{A}$ repeats the query of phase 1 subject to the restrictions defined in Game 1.
- **Guess:** Eventually, the adversary $\mathcal{A}$ outputs $b' \in \{0,1\}$.

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. The challenge ciphertext is a valid ciphertext under randomness $r = \frac{b}{c}$ and s.

Success probability: If $\mathcal{B}$ does not abort, $\mathcal{A}$'s view is identical to its view in the real attack. Now, we analyze the probability that $\mathcal{B}$ does not abort. We define three independent events:

ε_1: $\mathcal{B}$ does not abort as a result of any $\mathcal{A}$'s trapdoor queries, re-encryption key queries and re-encryption queries.

ε_2: $\mathcal{B}$ does not abort during the challenge phase.

We assume that $\mathcal{A}$ does not ask for the $\mathcal{O}_{H_1}$ of the same keyword twice. The probability that a $\mathcal{O}_{H_1}$ query to abort is $1/(q_{H_1})$. Assume that $\mathcal{A}$ makes at most q_{H_1} $\mathcal{O}_{H_1}$ queries, we have $pr[\varepsilon_1] \geq (1 - 1/(q_{H_1} + 1))^{q_{H_1}} \geq 1/e$ and $pr[\varepsilon_2] \geq 1/q_{H_1}$. Therefore $pr[\varepsilon_1 \wedge \varepsilon_2] \geq 1/(e \cdot q_{H_1})$. Since $\mathcal{B}$ does not abort with probability at least $1/(e \cdot q_{H_1})$, that $\mathcal{B}$'s success probability overall is at least $\epsilon/(e \cdot q_{H_1})$. So that $\mathcal{B}$ can use $\mathcal{A}$ to solve *mDBDH* problem with probability $(\frac{1}{2} + \frac{\epsilon}{e \cdot q_{H_1}})$. □

We now prove that our condition-hiding re-encryption with keyword search system is *IND-CCA* secure under the q-wDBDHI assumption.

Theorem 2: Our construction is IND-CCA secure in the random oracle model if the q-wDBDHI assumption holds in $\mathbb{G}$ *and the one-time signature Sig is strongly unforgeable.*

Proof: Suppose there exists a polynomial-time adversary $\mathcal{A}$ that breaks the *IND-CCA* security of system described above with advantage ϵ. We construct an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve q-wDBDHI problem with probability $\frac{\epsilon}{n}$ where n is the number of honest users. $\mathcal{B}$ takes as input a random q-wDBDHI challenge $(g, g_1, g_2, \ldots, g_q, g', Z)$, where Z is either $e(g, g')^{\frac{1}{\alpha}}$ or random element of $\mathbb{G}_T$. $\mathcal{B}$ interacts with $\mathcal{A}$ in the security game as follows:

- **Setup:** $\mathcal{B}$ generates a random polynomial $f(x) \in \mathbb{Z}_p[x]$ of degree $q-1$ and sets $h = g^{f(\alpha)} = \prod_{i=0}^{q-1} g_i^{f_i}$, where $(f_i, i \in \{0, 1, \ldots, q-1\})$ are the coefficients of the polynomial $f(x)$, h can be computed from $(g, g_1, \ldots, g_q)$. $\mathcal{B}$ selects a strongly unforgeable one-time signature $Sig = (\mathcal{G}, \mathcal{S}, \mathcal{V})$, $\mathcal{G}(\lambda) \rightarrow (ssk^*, svk^*)$ and $H_1 : \{0,1\}^* \rightarrow \mathbb{G}$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$. Let $f = g^{k_1}, u = g^{\gamma_1}, v = g^{-\gamma_1 \cdot svk^*} \cdot g_1^{\gamma_2}$, where $k_1, \gamma_1, \gamma_2 \in \mathbb{Z}_p^*$. The system public parameters are $(\mathbb{G}, \mathbb{G}_T, e, g, f, h, u, v, Sig, H_1, H_2)$.
  Hash Oracles: $\mathcal{B}$ builds the following oracles. $\mathcal{O}_{H_1}$ On input w to hash function H_1, $\mathcal{B}$ first checks whether tuple (w, h_1, r_1) occurs in table T_{H_1}. If yes, $\mathcal{B}$ responds $\mathcal{A}$ with r_1. Otherwise, $\mathcal{B}$ randomly pick $r_1 \in \mathbb{Z}_p^*$ and set $h_1 = g^{r_1}$. Finally, $\mathcal{B}$ responds $\mathcal{A}$ and records tuple (w, h_1, r_1) in table T_{H_1}. Similarly, for $\mathcal{O}_{H_2}$, $\mathcal{B}$ responds to a query for $H_2(ID)$ by picking a new random value $r_2 \in \mathbb{Z}_p^*$ for each new ID and sets $H_2(ID) = r_2$, and records tuple (ID, r_2) in table T_{H_2}.
- **Phase 1:** The target user's public/private key pair are set as, for random $y \in \mathbb{Z}_p^*$

$$pk_{i^*} = (X_{i^*}, Y_{i^*}) = (g_1, h^{y \cdot \alpha}),$$
$$sk_i = (\alpha, y \cdot \alpha) \textit{ which is unknown.}$$

  On $\mathcal{A}'s$ queries, $\mathcal{B}$ builds the following oracles.

  $\mathcal{O}_{pk}$: On input an identity ID_i, $\mathcal{B}$ picks random $x_i, y_i \in \mathbb{Z}_p^*$. $\mathcal{B}$ sets $X_i = g^{x_i}, Y_i = h^{y_i}$ and records the tuple (X_i, Y_i, x_i, y_i) in table T_k, and responds $\mathcal{A}$ with $pk_i = (X_i, Y_i)$.

  $\mathcal{O}_{sk}$: On input an identity ID_i, $\mathcal{B}$ picks random $x_i, y_i \in \mathbb{Z}_p^*$. $\mathcal{B}$ sets $X_i = g^{x_i}, Y_i = h^{y_i}$ and records the tuple (X_i, Y_i, x_i, y_i) in table T_k, and responds $\mathcal{A}$ with $(pk_i = (X_i, Y_i), sk_i = (x_i, y_i))$.

  $\mathcal{O}_{td}$: On input (ID_i, w), $\mathcal{B}$ responds $\mathcal{A}$ with $(g^{r_1})^{\frac{1}{y_i}}$.

  $\mathcal{O}_{rk}$: On input (ID_i, ID_j, w). $\mathcal{B}$ queries $\mathcal{O}_{td}((ID_i, w))$ to get rk_2. Let $F_{r_2}(x)$ be the $q-2$ degree polynomial $F_{r_2}(x) = f(x)/(x - r_2) = \prod_{i=0}^{q-2}(\mu_i x^i)$ where r_2 is the corresponding value to ID_i in table T_{H_2} and $(\mu_i, i = 0, 1, \ldots, q-2)$ are the coefficients of polynomial of $F_{r_2}(x)$. When $i = i^*$, $\mathcal{B}$ computes

$$rk_1 = Y_j^{\frac{1}{x_{i^*} - r_2}} = g^{\frac{y_j \cdot f(\alpha)}{\alpha - r_2}} = (\prod_{i=0}^{q-2} g_i^{\mu_i})^{y_i},$$
$$rk_3 = e(h, H_1(w))^{\frac{1}{x_{i^*} - r_2}} = e(g^{r_1}, \prod_{i=0}^{q-2} g_i^{\mu_i})$$

  When $i \neq i^*$, $\mathcal{B}$ can easily compute re-encryption key by using the known private key.

  $\mathcal{O}_{re}$: On input identities of the delegator and delegatee and the original ciphertext

$$\langle ID_i, ID_j, CT_i = (svk, C_1, C_2, C_3, C_4, C_5, C_6, C_7, \sigma)\rangle$$

  $\mathcal{B}$ queries $\mathcal{O}_{rk}(ID_i, ID_j, w)$ to get rk_1 and tuple (ID, r_2) in table T_{H_2}, and picks random $t \in \mathbb{Z}_p^*$ and computes

$$\langle C_1' = rk_1^t, C_2' = (X_i \cdot f^{-r_2})^{\frac{1}{t}}, C_3' = C_4^{\frac{1}{t}}, C_4' = C_4, C_5' = rk_3^t, C_6' = C_6, C_7' = C_7\rangle$$

  $\mathcal{B}$ responds $\mathcal{A}$ with

$$CT_j = (svk, C_1', C_2', C_3', C_4', C_5', C_6', C_7', \sigma)$$

  $\mathcal{O}_{dec}$: On input a ciphertext

$$CT_i = (svk, C_1, C_2, C_3, C_4, C_5, C_6, C_7, \sigma)$$

  by $\mathcal{A}$, if $svk = svk^*$, $\mathcal{B}$ terminates and returns a random bit. If $svk \neq svk^*$, when $i = i^*$, $\mathcal{B}$ computes

$$\left(\frac{e(C_6, h)}{e(C_1, C_3)^{\frac{\gamma_2}{y}}}\right)^{\frac{1}{\gamma_1 (svk - svk^*)}} = e(f, h)^s$$

  Then $\mathcal{B}$ computes $m = C_7 \cdot e(f, h)^s$. When $i \neq i^*$, $\mathcal{B}$ can easily compute m by using the known private key.
- **Challenge:** Eventually adversary $\mathcal{A}$ produces a pair of keywords m_0 and m_1, a condition w and a public key pk^* on which it wishes to be challenged. $\mathcal{B}$ chooses a random number $b \in \{0, 1\}$ such that $coin_w = 0$ and selects random $c \in \mathbb{Z}_p^*$ and responds $\mathcal{A}$ with the challenge ciphertext CT_b^*:

$$\langle C_1^* = h^{y \cdot c} = h^{y \cdot \alpha \cdot \frac{c}{\alpha}} = Y_{i^*}^r,$$

TABLE 2. Functionality comparison.

	F1	F2	F3	F4	F5
[4]	√	×	×	×	×
[5]	√	×	×	×	×
[12]	√	×	√	√	√
[21]	√	×	×	×	×
[31]	√	√	×	√	×
[32]	√	×	×	×	√
[36]	√	×	×	×	×
[37]	√	×	×	√	√
[46]	√	×	×	×	√
[49]	√	√	×	×	×
DSAS	√	√	√	√	√

Note: F1: Uni-Directional. F2: Proxy-Invisible. F3: Condition-Hiding. F4: Collusion-Resistance. F5: Keyword Search.

$$
\begin{aligned}
C_2^* &= (g_1)^{\frac{k_1}{c}} = f^{\frac{\alpha}{c}} = f^{\frac{1}{r}},\\
C_3^* &= g'^{\frac{1}{c}} = g^{\frac{\beta}{\alpha}\cdot\frac{\alpha}{c}} = g^{\frac{s}{r}},\\
C_4^* &= g' \cdot (g'^{k_1})^{r_2^*},\\
C_5^* &= e(\prod_{i=1}^{q} g_i^{f_{i-1}}, (g^{r_1}))^{\frac{1}{c}}\\
C_6^* &= g'^{\gamma_2},\\
C_7^* &= m_b \cdot e(g', \prod_{i=0}^{q-2} g_i^{f_{i+1}}) \cdot Z^{f_0}\\
\sigma^* &= \mathcal{S}(ssk^*, (C_6^*, C_7^*))\rangle
\end{aligned}
$$

where r_2* is the corresponding value to ID_{i^*} in table T_{H_2} and y_i is the corresponding value to pk^* in table T_k.

- **Phase 2:** $\mathcal{A}$ repeats the query of phase 1 subject to the restrictions defined in Game 2.
- **Guess:** Eventually, the adversary $\mathcal{A}$ outputs $b' \in \{0, 1\}$.

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 $s = \frac{\beta}{\alpha}$,$r = \frac{c}{\alpha}$ where $\beta = log_g g'$. If $Z = e(g, g')^{\frac{1}{\alpha}}$, CT_b^* is a valid ciphertext.

Success probability: If $Z = e(g, g')^{\frac{1}{\alpha}}$, $\mathcal{A}$'s view is identical to the view in the real attack environment. In contrast, $Z \neq e(g, g')^{\frac{1}{\alpha}}$, Z has a random distribution on $\mathbb{G}_T$, thus $\mathcal{A}$ cannot guess b with probability better than $1/2$. $\mathcal{A}$ finally output b', which used by $\mathcal{B}$ in its own game. If $b = b'$ then $\mathcal{B}$ decides that $Z = e(g, g')^{\frac{1}{\alpha}}$, otherwise, $\mathcal{B}$ decides that $Z \neq e(g, g')^{\frac{1}{\alpha}}$. Therefore, we have

$$
\begin{aligned}
|\, pr[\mathcal{B}(g, g_1, \ldots, g_q, g', e(g, g')^{1/\alpha}) = 0]\\
-pr[\mathcal{B}(g, g_1, \ldots, g_q, g', R) = 0]\, | \geq \epsilon/n.
\end{aligned}
$$

□

V. FUNCTIONALITY ANALYSIS

In this section, we compare our proposed DSAS scheme with other proxy re-encryption schemes of BVS [4], [5], HYF [21], SYL [31], WLQ [36], and ZCR [49], proxy searchable re-encryption schemes of FSGW [12], SCLL [32], WHYLW [37] and YM [46] in terms of functionality. Table 2 gives the comparison between our scheme and several related works in terms of features (i.e. uni-directional, proxy-invisible, condition-hiding, collusion-resistance, keyword search.)

- *Uni-directional*: Generally speaking, uni-directional proxy re-encryption is more superior than multi-directional proxy re-encryption, because the latter can be constructed by two different directions of the former, while the former can not be constructed by the latter. As shown in Table 2, all the schemes meet this security need.
- *Proxy-invisible*: As shown in Table 2, it is obvious that only [31] and our scheme have this property, which is more secure for e-healthcare system.
- *Condition-Hiding*: With condition-hiding, the cloud server get less sensitive information, which makes the system more secure. This expected goal is achieved in [12] and our scheme as shown in Table 2.
- *Collusion-resistance*: As shown in Table 2, it is obvious that only [12], [31] and our scheme achieve this security property where a delegator's private key is still secure even a dishonest cloud colludes with the delegatee.
- *Keyword Search*: As shown in Table 2, only [12], [32], [36], [37] and our scheme provide keyword search for e-healthcare, which makes ciphertext data more operational.

VI. PERFORMANCE ANALYSIS

In this section, we evaluate the performance of our DSAS system based on both real experiments and simulation.

A. EXPERIMENTAL SETTING

By adopting the Type A curves within the Paring Based Cryptography (PBC) library [22], we perform our proposed scheme on a laptop with 1.8-GHz Intel Core processor i5-8250U (Window 10 operation system, and a RAM of 8 GB) to act as the cloud server. This simulation environment is used to perform algorithms *ReEnc* and *Test*, which require a great computational and storage capability. In contrast, the users or sensor devices in our system require low computational capability, to perform algorithms *KeyGen*, *Enc*, *ReKeyGen*, *Trapdoor* and *Dec*, we deploy two Raspberry Pi sensor nodes (ARM Cortex-A53 1.2GHz 64-bit quad-core ARMv8 CPU) to form a wireless linked Industrial Internet of Things (IIoT). The nodes communicate with each other by ZigBee protocol. The sensor nodes communicate with the cloud server through one-hop or multihop manner. In the experiment, Let $|\mathbb{G}|$ denote a bit length of an element of $\mathbb{G}$, $|\mathbb{G}_T|$ denote a bit length of an element of $\mathbb{G}_T$. Since only schemes FSGW [12] and YM [46] are about conditional searchable proxy re-encryption, hence, we only compare our scheme with these two schemes, and the simulation results are exhibited in Fig. 4 to Fig. 11.

B. EXPERIMENTAL EVALUATION

In key generation phase, the system constructs public-private key pairs for each user with only 2 exponential operations

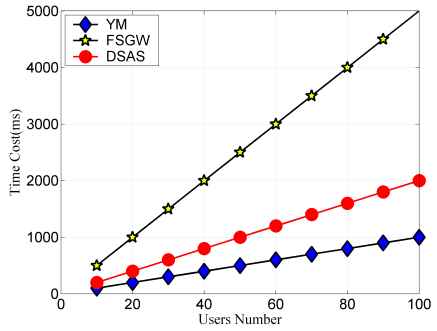

FIGURE 4. Performance of KeyGen.

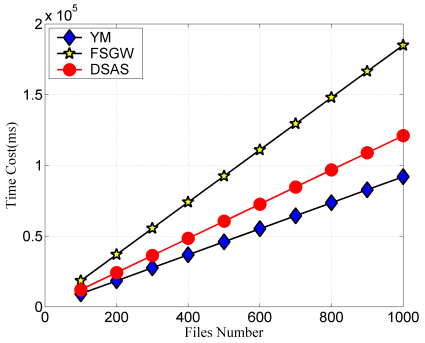

FIGURE 5. Performance of encrypt.

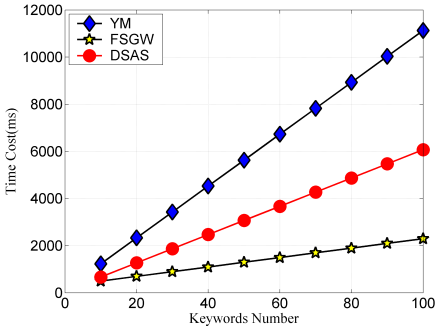

FIGURE 6. Performance of index encrypt.

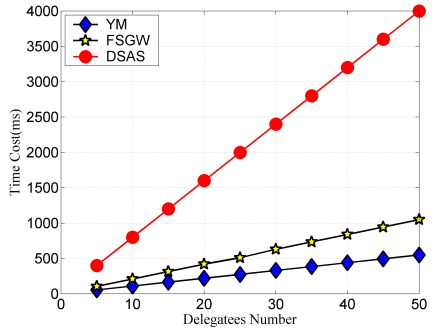

FIGURE 7. Performance of ReKeyGen.

in DSAS. Cost in FSGW [12] and YM [46] are 4 exponential operations and 1 exponential operation respectively. The computational cost of key generation in all schemes for each user is constant, which have significantly efficiency for lightweight devices in e-healthcare system as shown from Fig. 4, the computation cost increases linearly with the growth of users.

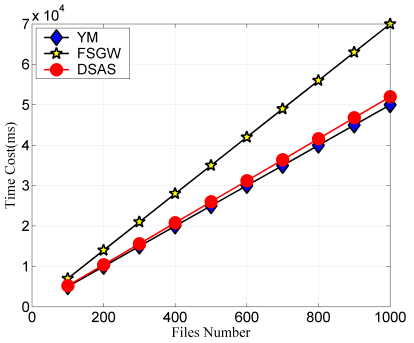

FIGURE 8. Performance of ReEnc.

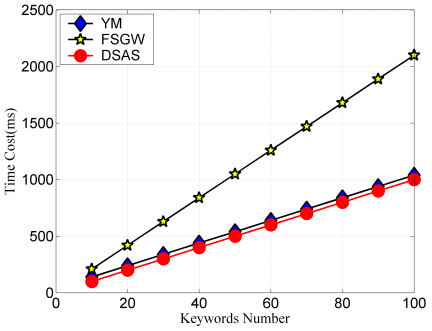

FIGURE 9. Performance of trapdoor.

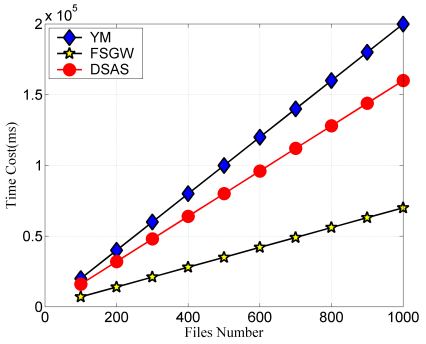

FIGURE 10. Performance of search.

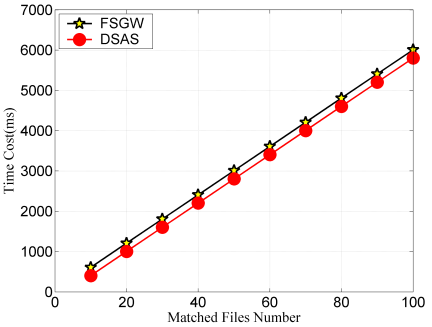

FIGURE 11. Performance of decrypt.

On receiving public-private key pairs, sensors for data collection encrypt the collected PHRs with doctor Alice's public key before uploading them to the cloud server. First, the sensors continuously collect PHRs F from physical envi-

ronments, then extract keyword w from these data. Second, run algorithm *Enc* to generate searchable index healthcare ciphertext under doctor Alice's public key pk_{Alice}. Finally, upload all ciphertext CT_{Alice} to the cloud server. There are $5|\mathbb{G}|+2|\mathbb{G}_T|$ cost for each file in DSAS, and $3|\mathbb{G}|+2|\mathbb{G}_T|$ and $3|\mathbb{G}|+1|\mathbb{G}_T|$ in FSGW [12] and YM [46] respectively. The computational cost of ciphertext generation in all schemes for each file is constant. Clearly from Fig. 5 and Fig. 6, the computation cost increases linearly with the growth of files.

On receiving the ciphertext, Alice is able to delegate search and decrypt operation to Bob through a cloud server if Alice is unavailable. The computational cost for re-encryption key generation is exhibited in Fig. 7. Cost in FSGW [12] and YM [46] are $4|\mathbb{G}|$ and $2|\mathbb{G}|$ respectively. DSAS requires $1|\mathbb{G}|+2|\mathbb{G}_T|$ for each re-encryption key, however, because DSAS considers embeds the trapdoor into re-encryption key, in which way can hidden the proxy condition. Obliviously, DSAS sacrifices some computing efficiency, thus obtaining better security.

Given re-encryption key, the cloud server runs algorithm *ReEnc* to convert the corresponding ciphertext to CT_{Bob} under Bob's public key with only 4 exponential operations in DSAS. Cost in FSGW [12] and YM [46] are 3 exponential operations and 6 exponential operation respectively. Next is about the encrypted keyword search, given trapdoor generated by Bob, the cloud server runs algorithm *Test* to perform information retrieve over encrypted PHRs. Cost of DSAS is only 3 pairing operations and cost in FSGW [12] and YM [46] are 2 pairing operations and 2 exponential operations and 3 pairing operations and 2 exponential operations respectively. As shown in Fig. 8 and Fig. 10, DSAS and YM [46] have more advantages on re-encryption and encrypted keyword search over FSGW [12].

Bob is able to search the converted ciphertext by running algorithm *Trapdoor* to generate a trapdoor t_w under keyword w and his private key sk_{Bob}. Cost in YM [46] and DSAS are 1 exponential operation and 4 exponential operations in FSGW [12]. Given the trapdoor t_w and ciphertext CT_{Bob}, the cloud server runs algorithm *Test* to find the matching ciphertext. Finally, Bob obtains the intended data by decrypting the matched ciphertext with his private key sk_{Bob} by running algorithm *Dec*. Because YM [46] has no capability for decryption, it is not considered in our comparison. As shown in Fig. 9 and Fig. 11, DSAS is as efficient in decryption as FSGW [12].

In summary, compared with FSGW, DSAS requires a little bit high cost in KeyGen and Encrypt; compared with YM, DSAS requires a little bit high cost in Index Encrypt. However, these results are acceptable since these costs are one-time, that is, users only need to take the corresponding costs when joining the system and uploading the e-healthcare records. In order to protect the privacy of conditions, we explore the embedding technology of trapdoor in searchable encryption, which make the performance of ReKeyGen unsatisfactory, which is what we need to improve in the future. Last but not least, our proposed scheme DSAS enjoys a good efficiency in encrypted information retrieve and ciphertext decryption requirement which show that our scheme DSAS is suitable for the e-healthcare system.

VII. CONCLUSION

In this paper, we presented a proxy-invisible condition-hiding proxy re-encryption scheme which supports keyword search that can be applied to securing data sharing and delegation in e-healthcare systems. With our new system, a doctor, Alice (delegator), may construct a conditional authorization for a doctor, Bob (delegatee), by specifying a re-encryption key. With the re-encryption key, the cloud server can perform ciphertext transformation so that Bob is able to access the PHRs original encrypted under Alice's public key, thus enabling secure delegation. The cloud server can operate search over encrypted PHRs on behalf of the doctor without learning information about the keyword or the underlying condition. Specifically, we achieved the property of proxy-invisible in the system. We have also obtained the property of collusion-resistance in the system, where a delegator's (Alice) private key is still secure even a dishonest cloud server colludes with the delegatee (Bob). We have demonstrated security through a rigorous proof, and the performance analysis confirms that our proposed scheme DSAS is efficient and practical.

REFERENCES

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 2005, pp. 205–222.

[2] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inf. Syst. Secur.*, vol. 9, no. 1, pp. 1–30, 2006.

[3] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *Proc. Int. Conf. Comput. Sci. Appl. (ICCSA)*, 2008, pp. 1249–1259.

[4] T. Bhatia, A. K. Verma, and G. Sharma, "Towards a secure incremental proxy re-encryption for e-healthcare data sharing in mobile cloud computing," *Concurrency Comput., Pract. Exper.*, vol. 32, no. 5, p. e5520, Mar. 2020.

[5] T. Bhatia, A. K. Verma, and G. Sharma, "Secure sharing of mobile personal healthcare records using certificateless proxy re-encryption in cloud," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 6, p. e3309, Jun. 2018.

[6] I. F. Blake, G. Seroussi, and N. Smart, "*Advances in Elliptic Curve Cryptography* (London Mathematical Society Lecture Note Series (317)), vol. 19. Cambridge, U.K.: Cambridge Univ. Press, no. 20, 2005, p. 666.

[7] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Advances in Cryptology-EUROCRYPT*. Berlin, Germany: Springer, 1998, pp. 127–144.

[8] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2004, pp. 506–522.

[9] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2007, pp. 535–554.

[10] H. Fang, X. Wang, and L. Hanzo, "Learning-aided physical layer authentication as an intelligent process," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2260–2273, Mar. 2019.

[11] H. Fang, L. Xu, and X. Wang, "Coordinated multiple-relays based physical-layer security improvement: A single-leader multiple-followers Stackelberg game scheme," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 197–209, Jan. 2018.

[12] L. Fang, W. Susilo, C. Ge, and J. Wang, "Chosen-ciphertext secure anonymous conditional proxy re-encryption with keyword search," *Theor. Comput. Sci.*, vol. 462, pp. 39–58, Nov. 2012.
[13] L. Fang, J. Wang, C. Ge, and Y. Ren, "Fuzzy conditional proxy re-encryption," *Sci. China Inf. Sci.*, vol. 56, no. 5, pp. 1–13, May 2013.
[14] J. Feng, L. T. Yang, R. Zhang, W. Qiang, and J. Chen, "Privacy preserving high-order bi-Lanczos in cloud-fog computing for industrial applications," *IEEE Trans. Ind. Informat.*, early access, May 28, 2020, doi: 10.1109/TII.2020.2998086.
[15] J. Feng, L. T. Yang, Q. Zhu, and K.-K.-R. Choo, "Privacy-preserving tensor decomposition over encrypted data in a federated cloud environment," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 4, pp. 857–868, Jul. 2020.
[16] J.-S. Fu, Y. Liu, H.-C. Chao, B. K. Bhargava, and Z.-J. Zhang, "Secure data storage and searching for industrial IoT by integrating fog computing and cloud computing," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4519–4528, Oct. 2018.
[17] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *Applied Cryptography and Network Security*. Berlin, Germany: Springer, 2007, pp. 288–306.
[18] D. He, M. Ma, S. Zeadally, N. Kumar, and K. Liang, "Certificateless public key authenticated encryption with keyword search for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3618–3627, Aug. 2018.
[19] Y. J. He, T. W. Chim, L. C. K. Hui, and S.-M. Yiu, "Non-transferable proxy re-encryption scheme for data dissemination control," *IACR Cryptol. ePrint Arch.*, vol. 2010, p. 192, Jan. 2010.
[20] Q. Huang, L. Wang, and Y. Yang, "Secure and privacy-preserving data sharing and collaboration in mobile healthcare social networks of smart cities," *Secur. Commun. Netw.*, vol. 2017, pp. 1–12, Aug. 2017.
[21] Q. Huang, Y. Yang, and J. Fu, "PRECISE: Identity-based private data sharing with conditional proxy re-encryption in online social networks," *Future Gener. Comput. Syst.*, vol. 86, pp. 1523–1533, Sep. 2018.
[22] B. Lynn. (2006). *PBC Library*. [Online]. Available: http://crypto.stanford.edu/pbc
[23] M. Ma, D. He, D. Kumar, K.-K. R. Choo, and J. Chen, "Certificateless searchable public key encryption scheme for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 759–767, May 2017.
[24] Y. Miao, J. Ma, X. Liu, F. Wei, Z. Liu, and X. A. Wang, "m2-ABKS: Attribute-based multi-keyword search over encrypted personal health records in multi-owner setting," *J. Med. Syst.*, vol. 40, no. 11, p. 246, Nov. 2016.
[25] M. Naz, F. A. Al-zahrani, R. Khalid, N. Javaid, A. M. Qamar, M. K. Afzal, and M. Shafiq, "A secure data sharing platform using blockchain and interplanetary file system," *Sustainability*, vol. 11, no. 24, p. 7054, 2019.
[26] 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 Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 94–105, May 2018.
[27] 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 Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 883–897, Sep./Oct. 2018.
[28] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, "White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1274–1288, Jun. 2015.
[29] S. Niu, L. Chen, J. Wang, and F. Yu, "Electronic health record sharing scheme with searchable attribute-based encryption on blockchain," *IEEE Access*, vol. 8, pp. 7195–7204, 2020.
[30] P. Xu, S. He, W. Wang, W. Susilo, and H. Jin, "Lightweight searchable public-key encryption for cloud-assisted wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3712–3723, Aug. 2018.
[31] J. W. Seo, D. H. Yum, and P. J. Lee, "Proxy-invisible CCA-secure type-based proxy re-encryption without random oracles," *Theor. Comput. Sci.*, vol. 491, pp. 83–93, Jun. 2013.
[32] J. Shao, Z. Cao, X. Liang, and H. Lin, "Proxy re-encryption with keyword search," *Inf. Sci.*, vol. 180, no. 13, pp. 2576–2587, 2010.
[33] S. Tahir, S. Ruj, Y. Rahulamathavan, M. Rajarajan, and C. Glackin, "A new secure and lightweight searchable encryption scheme over encrypted cloud data," *IEEE Trans. Emerg. Topics Comput.*, vol. 7, no. 4, pp. 530–544, Oct. 2019.
[34] H. Wang, X. Dong, Z. Cao, D. Li, and N. Cao, "Secure key-aggregation authorized searchable encryption," *Sci. China Inf. Sci.*, vol. 62, no. 3, p. 39111, Mar. 2019.
[35] H. Wang, J. Ning, X. Huang, G. Wei, G. S. Poh, and X. Liu, "Secure fine-grained encrypted keyword search for e-healthcare cloud," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1307–1319, May/Jun. 2019.
[36] Q. Wang, W. Li, and Z. Qin, "Proxy re-encryption in access control framework of information-centric networks," *IEEE Access*, vol. 7, pp. 48417–48429, 2019.
[37] X. A. Wang, X. Huang, X. Yang, L. Liu, and X. Wu, "Further observation on proxy re-encryption with keyword search," *J. Syst. Softw.*, vol. 85, no. 3, pp. 643–654, 2012.
[38] J. Weng, R. H. Deng, X. Ding, C.-K. Chu, and J. Lai, "Conditional proxy re-encryption secure against chosen-ciphertext attack," in *Proc. 4th Int. Symp. Inf., Comput., Commun. Secur. (ASIACCS)*, 2009, pp. 322–332.
[39] J. Weng, Y. Yang, Q. Tang, R. H. Deng, and F. Bao, "Efficient conditional proxy re-encryption with chosen-ciphertext security," in *Proc. Int. Conf. Inf. Secur.* Berlin, Germany: Springer, 2009, pp. 151–166.
[40] L. Xu, C. Xu, J. K. Liu, C. Zuo, and P. Zhang, "Building a dynamic searchable encrypted medical database for multi-client," *Inf. Sci.*, vol. 527, pp. 394–405, Jul. 2020.
[41] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin, "Conditional identity-based broadcast proxy re-encryption and its application to cloud email," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 66–79, Jan. 2016.
[42] S. Xu, Y. Li, R. H. Deng, Y. Zhang, X. Luo, and X. Liu, "Lightweight and expressive fine-grained access control for healthcare Internet-of-Things," *IEEE Trans. Cloud Comput.*, vol. 10, no. 1, pp. 474–490, Jan. 2022.
[43] S. Xu, G. Yang, and Y. Mu, "Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation," *Inf. Sci.*, vol. 479, pp. 116–134, Apr. 2019.
[44] S. Xu, G. Yang, Y. Mu, and R. H. Deng, "Secure fine-grained access control and data sharing for dynamic groups in the cloud," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 8, pp. 2101–2113, Aug. 2018.
[45] L. Yang, Q. Zheng, and X. Fan, "RSPP: A reliable, searchable and privacy-preserving e-healthcare system for cloud-assisted body area networks," in *Proc. IEEE Conf. Comput. Commun.*, May 2017, pp. 1–9.
[46] Y. Yang and M. Ma, "Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for e-health clouds," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 4, pp. 746–759, Apr. 2016.
[47] X. Yao, Y. Lin, Q. Liu, and J. Zhang, "Privacy-preserving search over encrypted personal health record in multi-source cloud," *IEEE Access*, vol. 6, pp. 3809–3823, 2018.
[48] W. A. Yasnoff, "A secure and efficiently searchable health information architecture," *J. Biomed. Informat.*, vol. 61, pp. 237–246, Jun. 2016.
[49] P. Zeng and K.-K. R. Choo, "A new kind of conditional proxy re-encryption for secure cloud storage," *IEEE Access*, vol. 6, pp. 70017–70024, 2018.
[50] R. Zhang, R. Xue, and L. Liu, "Searchable encryption for healthcare clouds: A survey," *IEEE Trans. Services Comput.*, vol. 11, no. 6, pp. 978–996, Nov./Dec. 2017.
[51] R. Zhou, X. Zhang, X. Du, X. Wang, G. Yang, and M. Guizani, "File-centric multi-key aggregate keyword searchable encryption for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3648–3658, Aug. 2018.

LINLIN XUE received the B.E. degree in electronic information engineering and the Ph.D. degree in electromagnetic field and microwave technology from the University of Science and Technology of China, Anhui, China, in 2008 and 2013, respectively. She was a Lecturer at the Zhejiang University of Technology, from 2013 to 2019. Since 2019, she has been a Lecturer with the Zhejiang University of Science and Technology. She has been authored and coauthored over 20 journal and conference papers in her areas of expertise. Her current research interests include optical communications and searchable encryption.