The ISC Int'l Journal of Information Security

July 2020, Volume 12, Number 2 (pp. 129–140)

http://www.isecure-journal.org

Attribute-based Access Control for Cloud-based Electronic Health Record (EHR) Systems

Maryam Zarezadeh [1], Maede Ashouri Taluki [1,*], and Mohammad Siavashi [2]

[1] *Department of Information Technology Engineering, University of Isfahan, Isfahan, Iran*

[2] *Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran*

ARTICLE INFO.

Article history:
Received: 7 March 2019
Revised: 16 March 2020
Accepted: 27 May 2020
Published Online: 31 May 2020

Keywords:
Access Control, Electronic Health Record, Attribute-based Encryption, EHR, Cloud Storage.

Abstract

The electronic health record (EHR) system facilitates integrating patients' medical information and improves service productivity. However, user access to patient data in a privacy-preserving manner is still a challenging problem. Many studies concerned with security and privacy in EHR systems. Rezaeibagha and Mu [1] have proposed a hybrid architecture for privacy-preserving accessing patient records in a cloud system. In their scheme, encrypted EHRs are stored in multiple clouds to provide scalability and privacy. In addition, they considered a role-based access control (RBAC) such that for any user, an EHR access policy must be determined. They also encrypt the EHRs by the public keys of all users. So, for a large amount of EHRs, this scheme is not efficient. Furthermore, using RBAC for access policy makes the policy changing difficult. In their scheme, users cannot search on encrypted EHRs based on diseases, and some physicians must participate in the data retrieval by a requester physician. In this paper, we address these problems by considering ciphertext-policy attribute-based encryption (CP-ABE), which is conceptually closer to the traditional access control methods such as RBAC. Our secure scheme can retrieve encrypted EHR based on a specific disease. Furthermore, the proposed scheme guarantees the user access control and the anonymity of the user or data owner during data retrieval. Moreover, our scheme is resistant against collusion between unauthorized retrievers to access the data. The analysis shows that our scheme is secure and efficient for cloud-based EHRs.

1 Introduction

The development of cloud computing provides data sharing between people through data storage services. Data sharing in electronic health record (EHR) systems is significant to improve the quality of healthcare records and disease diagnostics. However, the security and privacy of users should be preserved since the patient's information may be exploited. So far, different encryption schemes to protect user's confidential data in the cloud storage are proposed, such as [2–4]. Rezaeibagha and Mu [1] also presented an access control scheme for the EHR system; we named the scheme as distributed clinical data sharing (DCDS) scheme. In this scheme, communication and data sharing between private domain entities,

* Corresponding author.
Email addresses: m.zarezadeh@eng.ui.ac.ir, m.ashouri@eng.ui.ac.ir, m.siavashi@cse.shirazu.ac.ir

ISSN: 2008-2045

including hospitals, private clinics, and emergency departments, are provided. In the DCDS scheme [1], the hybrid cloud structure (public and private) is used. The EHRs stored in public could be used by users such as medical researchers, insurance companies. Physicians in the private cloud have access to data if the threshold number of authorized physicians are present. This restriction increases the privacy of the patient's records. The access control mechanisms are different in these two clouds. In the private cloud, secret sharing and role-based access control (RBAC) mechanisms are used while the access control mechanism in the public cloud is only based on RBAC.

Each policy is defined as a tuple of roles, objects, policies, domain, time, and objective. One server is responsible for system setup, key generation, and key management. This server selects the public/private key pair and computes public and private keys for any user who sends his identity to the server and receives his private key. In this mechanism, when the public cloud server sends a request to transfer the patient's record, policy transformation is performed. The server of the private cloud retrieves the encrypted data of a patient and re-encrypts it with another key, which is based on the secret key of a requester user. The policy transformation is based on ciphertext policy attribute-based encryption (CP-ABE), which can be used to determine the access structure of stored EHR with RBAC policy. In the scheme of Rezaeibagha and Mu [1], an attribute-based proxy re-encryption [2], which is a type of CP-ABE is considered.

In the DCDS scheme [1], there are some weaknesses. A server encrypts any EHR which can be decrypted by the key of any user. Hence, the EHRs must be encrypted by the keys of all authorized users. Also, in the DCDS scheme [1], the access policy for each outsourced EHR and any user is specified. So, the size of the access table is increased by the growth of the number of authorized users. In this paper, we use the attribute-based encryption to overcome these weaknesses and present an efficient access control scheme. In our scheme, a data owner outsources his encrypted data to a cloud, and only users with attributes satisfying the defined access policy can retrieve data. Furthermore, a proxy server helps the users for decrypting the searched data.

The rest of this paper is organized as follows. Section 2 summarizes the related works regarding attribute-based access control. The DCDS scheme [1] will be described in Section 3. Section 4 analyzes their scheme. In Section 5, we propose a scheme addressing the problems of the DCDS scheme [1]. Section 6 analyzes the performance of the proposed scheme; security analysis of our scheme is described in Section 7. Finally, Section 8 concludes the paper.

2 Related Work

In this section, we briefly review some researches on the issue of EHRs sharing based on attribute-based encryption. Bonaloh *et al.* [3] proposed an efficient system that facilities patients to share partial access rights and search over records in the public-key setting. However, in traditional public key infrastructure, only a single user can access the EHRs, which leads to limited efficiency. Besides, the anonymity of users cannot be provided. To improve this issue, some researches applied ABE to provide security and fine-grained access control for the outsourced data.

2.1 Centralized Schemes

Narayan *et al.* [4] suggested an infrastructure for EHR systems in which EHRs are encrypted with a type of CP-ABE. They supposed that the cloud prepares reliable storage for data, and the cloud provider can see or copy a stored data. Users store their encrypted data in the cloud; data access is provided based on the identity information of the requester. However, the proposed method incurs a high computational cost, and the length of ciphertexts increases with the number of users. Furthermore, Alshehri *et al.* [5] proposed a scheme for a secure cloud-based EHR system. Their technique uses CP-ABE that is based on elliptic curve cryptography and bilinear maps. Encrypting EHRs depends on healthcare providers while decrypting them needs attribute set for specific access. Also, Wang *et al.* [6] suggested an EHR sharing model in the cloud using CP-ABE, which supports privacy and fine-grained access control. But, these two works only provide data confidentiality and do not consider the unforgeability and the legitimacy of EHRs.

To implement a dynamic access policy for EHRs in clouds, Liu *et al.* [7] presented a hierarchical comparison-based encryption scheme [8] that incorporates an attribute hierarchy into the comparison-based encryption. To improve the performance of the encryption, the proposed scheme encrypts a ciphertext with a few generalized attributes at a higher level rather than several attributes at a lower level. They developed a dynamic policy updating scheme by applying the proxy re-encryption method, which has been made by them. This scheme avoids the transmission of ciphertext and minimizes the computational overhead of a data owner by delegating the policy updating operations to the cloud. Moreover, Cai *et al.* [9] implemented the electronic health records in the mobile health cloud by considering privacy and efficiency. Their scheme is proven secure in

the random oracle model under the static decisional bilinear Diffie-Hellman assumption.

Authors in [10] presented a fine-grained attribute-based access control for health records. The proposed scheme provides shared information by the common access sub-policy based on different patients' access policies. The scheme combines the encryption of health records under the common access sub-policy for reducing the time consumption of encryption and decryption. The scheme of [10] guarantees preserving data privacy; however, it is impractical for health records due to computational complexity and scalability issues. Furthermore, Joshi *et al.* [11] proposed a mechanism that uses ABE and provides delegated access to EHRs. In their approach, a complex knowledge graph for detailing the roles and attributes of the medical organization's stakeholders and the relationships between them is designed. This approach transfers the EHR management overhead from the patient to the medical organization and provides delegation of cloud-based EHR access authority for the medical providers.

Tao *et al.* [12] suggested a scheme for secure access to personal health records where a group-oriented CP-ABE classifying users into different groups is used. The users with an identical identifier can combine their attributes to complete the decryption. Finally, the decryption operation is completed when the union of their attributes satisfies the access control policy. Zhang *et al.* [13] investigated that most of the proposed schemes have two problems. First problem is that the proposed schemes do not support large attribute universe and hence their practicality is limited. Other problem that due to the access policy is embedded in ciphertext, the cost of decryption is high. They introduced a method called linear secret sharing with multiple values to improve the expressiveness of access policy. In this scheme, each attribute is divided into two parts namely the attribute name and its value. Hence, their scheme hides sensitive attribute values and privacy of users in health records is preserved. Also, Rezaeibagha and Mu [1] suggested a secure EHR system architecture for secure data sharing based on secret sharing and RBAC to preserve patients' privacy; we termed it as the DCDS scheme. To better manage the system, they suggested that the EHRs are stored in different types of clouds, i.e. a public cloud and a private cloud. However, this scheme has problems for data storing and data retrieving in the clouds.

2.2 Decentralized Schemes

Huang *et al.* [14] proposed a scalable sharing and access framework for EHRs stored on cloud. They considered two types of ABE schemes: key-policy ABE (KP-ABE) and CP-ABE to apply the fine-grained access control. In their proposed scheme, a data owner can select the users which he would like to share his EHR with them. But this approach needs key management and computational overhead. Li *et al.* [15] suggested a framework for fine-grained access control of EHRs. They adopted the concept of multi-authority ABE for multi-owner settings but their method suffers from the key escrow problem. Then, the authors [16] for solving this issue, presented an enhancement to the multi-authority ABE framework. Li *et al.* [16] proposed a patient-centric framework and presented methods for access control to EHRs. EHRs are distributed in semi-trusted servers and are encrypted with ABE. Their method guarantees a high degree of privacy by using multi-authority ABE although only the KP-ABE systems can use this method. The methods [15, 16] did not preserve the privacy of EHRs in the situation of a medical emergency. Ermakova and Fabian [17] presented a new architecture for sharing EHRs in a multi-cloud environment. The proposed architecture applies ABE and uses secret sharing to split the encrypted EHR into shares stored at several cloud providers. The problem of their method [17] is data management where the EHR processed and shared in multiple cloud providers requires suitable and secure management.

Liu *et al.* [18] adopted an attribute-based signature as the signature part allowing a party, who possesses the attribute set satisfying the access policy, to sign his records with his secret key. They used the CP-ABE as the encryption construction and their scheme provides confidentiality, authenticity, unforgeability, anonymity and collusion resistance. Li *et al.* [19] presented a decentralized key-policy ABE scheme and built up an EHR system providing fine-grained access policy to be extremely expressive and ciphertext to be maintained at a constant level. Furthermore, Charanya *et al.* [20] suggested an ABE scheme for secure sharing of EHRs. They used the CP-ABE and considered the key updating problem for the situation where the policy is changed and the keys must be reconstructed. They solved this problem by applying a proxy re-encryption technique with one time pad pin and data is available only to authorized users. Ramu *et al.* [21] presented a modified CP-ABE scheme with user revocation to achieve a fine-grained access control of EHRs in a cloud system. They solved the key escrow problem using two authorities to generate users' keys. But their scheme does not generate a constant ciphertext size.

In the following, the DCDS scheme [1] is considered. We analyze this scheme and propose an improved scheme. As seen in the literature review, many

researchers investigated the attribute-based access control for an EHR system. We also use the ABE to encrypt data to address the problems of the DCDS scheme [1] and provide access to authorized users. Our scheme allows users to search for the specific disease on encrypted EHRs.

3 Review of Distributed Clinical Data Sharing (DCDS) Scheme

In this section, we briefly describe the DCDS scheme of Rezaeibagha and Mu [1]. The DCDS scheme [1] consists of a public cloud and a private cloud. The server in the public domain (S_{pub}) contains the outsourced EHRs which are accessed by public users such as researchers, government, insurance companies, etc. The server in the private domain (S_{pri}) stores EHRs for users of the private domain such as physicians or specialists. Hence, two different access control mechanisms are used in two clouds. The access control of the public cloud is based on RBAC, while RBAC and secret sharing (sh_n^t) are used in the private cloud. The DCDS scheme [1] is presented as follows. It is supposed that all messages are exchanged on a secure channel.

3.1 System Setup

Let S be a fully trusted entity by other participants, which sets up the EHR system. S selects a pair of public/private keys (PK, SK) and generates the keys of other participants using its keys.

3.2 User Setup

In this phase, a user U and a server S send the following messages to each other.

(1) $U \rightarrow S$: U, $REQ_U(id_i)$, $i \in \alpha$
(2) $S \rightarrow U$: SK_U

A user of public or private cloud sends a registration request to the server S in which α is a set of indices with respect to the EHR and $REQ_U\,(id_i)$ is a request from the user U to obtain the decryption key for a record with identity id_i. The server S calls the key generation algorithm with the role assigned to the user U and generates and returns the corresponding key SK_U.

3.3 Reading Patient Record

The following messages are exchanged between a physician $\mathcal{P}$ and S_{pri} in the private domain.

(1) $\mathcal{P} \rightarrow S_{pri}$: U, $REQ_{\mathcal{P}}(id_i)$, $i \in \alpha$
(2) $S_{pri} \rightarrow \mathcal{P}$: $\bar{o_i}$

The physician $\mathcal{P}$ sends a request to the server S_{pri} to read the record o_i with respect to id_i. Next, S_{pri} retrieves the encrypted record $\bar{o_i}$. The physician $\mathcal{P}$ can decrypt $\bar{o_i}$ with his private key $SK_{\mathcal{P}}$.

3.4 Reading Shared Patient Record

The following messages are exchanged between a set of physicians and the server S_{pri} in the private domain.

(1) $\{\mathcal{P}_j\}_1^t \rightarrow S_{pri}$: U, $REQ_{\mathcal{P}_j}(id_i)$, $i \in \alpha$
(2) $S_{pri} \rightarrow \{\mathcal{P}_j\}_1^t$: $sh_n^t(\bar{o_i})$
(3) $\{\mathcal{P}_j\}_1^t$ collaboratively compute o_i

A set of physicians $\{\mathcal{P}_j\}_1^t$ ask the server S_{pri} to access the shared EHR o_i. S_{pri} retrieves and sends the encrypted EHR $sh_n^t(\bar{o_i})$ to the applicant physicians such that at least t physicians must collaborate to retrieve o_i.

3.5 Reading a Record by Researcher

The following messages are exchanged between a researcher R and the public cloud server S_{pub}.

(1) $R \rightarrow S_{pub} : REQ_R\,(id_i)\,,\ i \in \alpha$
(2) $S_{pub} \rightarrow R :\ \widehat{o_i}$

The researcher R sends a request to S_{pub} to access the record o_i with respect to the identity id_i. S_{pub} searches the corresponding record $\widehat{o_i}$. The researcher R can read $\widehat{o_i}$ with his private key SK_R.

3.6 Updating Patient Record

The physician $\mathcal{P}$ can send a request to the server S_{pri} for updating the EHR o_i.

(1) $\mathcal{P} \rightarrow S_{pri} : o_i',\ i \in \alpha$
(2) S_{pub} : $\bar{o_i'}$ (stores $\bar{o_i'}$ and updates the access policy)

Suppose that the physician $\mathcal{P}$ has retrieved the EHR o_i and wants to update it. The physician $\mathcal{P}$ sends the updated EHR o_i' to server S_{pri}. Next, S_{pri} stores the encrypted of o_i', dented by $\bar{o_i'}$, in cloud storage.

3.7 Policy Transformation

Policy transformation makes possible the EHR transformation between two domains as follows.

(1) $S_{pub} \rightarrow S_{pri} : o_i,\ i \in \alpha$
(2) $S_{pri} \rightarrow S_{pub} : \widehat{o_i} = ABRE(key, \bar{o_i})$

S_{pub} requests to access the EHR o_i stored in the private cloud storage. S_{pri} retrieves the corresponding encrypted $\overline{o_i}$. Then, S_{pri} runs the attribute-based re-encryption $ABRE$ [2] by taking the key and $\overline{o_i}$ as input. Finally, the record $\widehat{o_i}$ with new access policy will be sent to the server S_{pub}.

4 Motivation

Rezaeibagha and Mu [1] suggested an access-control mechanism for an EHR system with the hybrid cloud structure, which allows handling users with different access privileges. However, their scheme provides data confidentiality by inefficient access control and does not support a mechanism for searching on encrypted EHRs. To be more precise, their scheme has the following security and efficiency weaknesses.

- In the system setup phase, a trusted authority generates the public/private keys for users. Then, a data owner encrypts each data with the public keys of all cloud users. This requirement decreases the applicability of the scheme for a large amount of EHRs.
- Access policy mechanism is based on RBAC in which permission to any EHR is specified in the access table. Hence, it makes the policy change difficult for adapting the EHR system. That is, for any policy update, the cloud server must search all access control records.
- A physician must collaborate with other physicians to read the shared record; i.e., he cannot read the record independently. This collaboration is time-consuming and increases time for access to a patient's record in emergency cases.
- A user of the EHR system sends the identity of an EHR to the server. This mechanism is not suitable for data outsourcing since the user has not the search capability. In other words, a user of a cloud server should be able to search the outsourced EHRs based on the disease in outsourced EHRs.

Therefore, in this paper, we address the limitations of Rezaeibagha and Mu's scheme [1] by presenting a scheme that supports access control and search capability on outsourced EHRs. In our scheme, a data owner, who wants to outsource his/her record, computes a parameter as an indicator that a cloud server will later use it to check whether a record is matched to the received query. Finally, the data owner stores the encrypted record and corresponding values on cloud storage. Later, a retriever generates a query for a record and sends it to the server. The server matches the query with the values received from the data owners. Then, the server determines whether the retriever can access the encrypted files based on his attribute set and the access tree. If the attribute set of the user satisfies the access tree, the server then sends the encrypted record to him/her. Next, the retriever decrypts the records with the help of a proxy server and retrieves the outsourced data. Our contributions can be described as follows.

- We focus on access control and enabling search over encrypted records, simultaneously. In the proposed scheme, only the users with the attributes satisfying the data owner-defined access policy can retrieve the encrypted data.
- In our scheme, a data owner encrypts the records only once, and there is no need to encrypt them by all the public keys of the retrievers. This method would facilitate data updating and data outsourcing.
- To reduce the computational cost of decryption, users can delegate most attribute-based decryption to the proxy server without disclosing the record.
- Our scheme provides user anonymity during data retrieval and resists against collusion among unauthorized users.

5 The Proposed Scheme

In this section, we suggest a scheme that addresses the mentioned problems of Rezaeibagha and Mu's scheme [1].

5.1 Preliminaries and System Architecture

In this section, we give a brief review of the definitions and describe the system architecture used in the proposed scheme.

5.1.1 Bilinear Map

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of prime order p and $\mathbb{G}_1$ and $\mathbb{G}_2$ denote an additive group and a multiplicative group, respectively. Let g be a generator of $\mathbb{G}_1$. A bilinear map is an injective function $e: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ with the following properties:

- Bilinearity: for all $u, v \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p$, $e\left(u^a, v^b\right) = e(u,v)^{ab}$.
- Non-degeneracy: $e(g,g) \neq 1$.
- Computability: for all $u, v \in \mathbb{G}_1$, $e(u,v)$ is efficiently computable.

5.1.2 Access Tree

Suppose that $\mathcal{T}$ denotes a tree and represents an access structure for access policy. Each non-leaf node of $\mathcal{T}$ is described by the number of its children nodes (num_x) and a threshold gate value (k_x) such that $0 < k_x \leq num_x$. When $k_x = 1$, the threshold gate is an 'OR' gate and when $k_x = num_x$, it is an 'AND' gate. So, to determine different access policies, k_x can be set to different values. For example k_x is set to 1, $num_x/2$ and num_x for read, write and delete permissions, respectively. Also, each leaf node x of the tree is described by an attribute ($attr_x$) and the threshold $k_x = 1$. In the access tree, any child node of node x has a label from 1 to num_x and $index(x)$

returns a label that is associated with it. Also, we consider that $parent(x)$ denotes the parent node of the node x.

5.1.3 Satisfying an Access Tree

Let $\mathcal{T}_x$ be a subtree of $\mathcal{T}$ rooted at node x and $\mathcal{T}_x(\gamma) = 1$ denotes a set of attributes γ satisfying the access tree $\mathcal{T}_x$. $\mathcal{T}_x(\gamma)$ is recursively computed as follows. If the node x is a leaf node, then $\mathcal{T}_x(\gamma) = 1$ if and only if $attr_x \in \gamma$. When x is a non-leaf node, $\mathcal{T}_{x'}(\gamma)$ must be evaluated for all children x' of node x. If and only if at least k_x children of node x return 1, then $\mathcal{T}_x(\gamma) = 1$.

5.1.4 System Architecture

In the proposed scheme, the access policy can be described by an access tree $\mathcal{T}$, where the interior nodes consist of 'AND' and 'OR' gates, and the leaves consist of different attributes, which have stronger expressiveness. Moreover, CP-ABE is conceptually closer to traditional access control methods such as RBAC. To describe the RBAC of Rezaeibagha and Mu's scheme [1], we use the CP-ABE scheme [22]. As shown in Figure 1, the system includes the following participants.

- **Trusted authority (TA).** It is a fully trusted participant who is responsible for generating the public parameters and the keys of users.
- **Data owner.** This is the entity that encrypts his data under the access control policy and uploads them to the cloud storage.
- **Retriever/user.** This entity wants to access the encrypted data. To reduce the decryption overhead, a proxy server participates in the decryption process of an authorized data for a retriever.
- **Public cloud server** (S_{pub})**.** A semi-honest participant that manages the public cloud. The public cloud provides storage for the cloud-based system, which can be used by public users such as researchers.
- **Private cloud server** (S_{pri})**.** A semi-honest participant that manages the private cloud. The private cloud contains the private data which can only be accessed by special users such as physicians and specialists.
- **Proxy server.** A proxy server can be deployed inside each enterprise. This semi-honest entity is introduced to help retrievers to decrypt the files and facilitates the usage of cloud service.

To enable access control and to retrieve outsourced data on cloud storage, we have the following goals: 1) Data confidentiality: the scheme prevents the public cloud server or the private cloud server from accessing the outsourced records. 2) Anonymity: the identity of the retriever or the data owner is kept secret from the proxy server or the public/private cloud servers. 3) Collusion resistance: the colluding retrievers cannot access the unauthorized files. Moreover, a proxy server and the servers of clouds cannot obtain the outsourced records by colluding with each other.

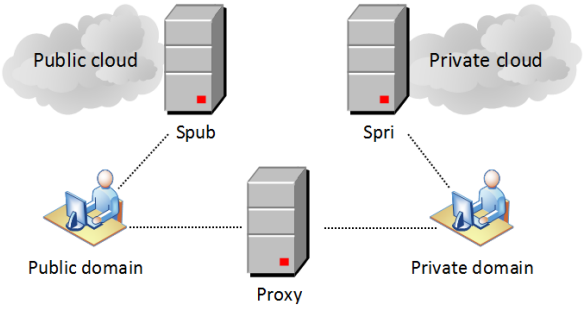


Figure 1. The system architecture

5.2 Threat Model

We assume that two cloud servers in public and private domains are semi-honest; in other words, they follow the proposed scheme while may try to obtain other additional information from the requests of retrievers. Also, we consider the proxy server as a semi-trusted server deployed to participate in retrievers in the decryption phase. It is supposed that the public cloud server, the private cloud server, and a proxy server will not collude with the unauthorized users for data retrieval. But, malicious users may collude to access the data which they do not have enough attributes to satisfy the access policy.

5.3 Construction

5.3.1 Setup

The TA chooses a bilinear group $\mathbb{G}$ of prime order p with generator g. Also, it selects two random exponents $\alpha,\ \beta\ \in \mathbb{Z}_p$ and a hash function $H: \{0,1\}^* \rightarrow \mathbb{G}$. The outputs of this phase are a master secret MK and a public parameter PK as follows.

$$MK = (\beta,\ g^{\alpha}),\ PK = (\mathbb{G},\ g, h = g^{\beta},\ \omega = e(g,g)^{\alpha})$$

5.3.2 Key Generation

In this phase, the TA generates the private key SK for a retriever U_i to decrypt the ciphertexts and the key A_o for a data owner. The TA selects a random $z_o \in \mathbb{Z}_p$ and returns $A_o = H(ID_o)^{z_o}$ to the data owner with identity ID_o. Also, for any retriever such as a physician or a researcher with identity ID_i, the TA selects a random $r \in \mathbb{Z}_p$ for each attribute $\lambda_j \in \Lambda_i$. This phase outputs the private key SK as

$$SK = (D = g^{(\alpha+r)/\beta},\ \{D_j = g^r H(\lambda_j)^{r_j}, D'_j = g^{r_j}\}_{\lambda_j \in \Lambda_i}).$$

It is notable that Λ_i denotes the attribute set belongs to the retriever U_i.

5.3.3 Encryption

Consider a data owner with the EHR about disease specified by d_i. For data outsourcing, the data owner with identity ID_o randomly selects $b \in \mathbb{Z}_p$ and computes the parameter $I_i = e(A_o, H(d_i)^b)$ as an indicator such that a cloud server can use it to know whether a data record matches to the query of a retriever. Also, the data owner generates pseudonym $P_o = {A_o}^b$. Then, he publishes P_o and runs the encryption algorithm. The encryption algorithm encrypts a record M under the access tree $\mathcal{T}$. The algorithm selects a polynomial q_x with degree $d_x = k_x - 1$, for each node x in the access tree $\mathcal{T}$. Then, a random $s \in \mathbb{Z}_p$ is selected and $q_R(0)$ is set to s for the root node R. The ciphertext is computed as follows, in which Y represents the set of leaf nodes in the access tree $\mathcal{T}$.

$$CT = (\mathcal{T}, P_o, \tilde{C} = M\omega^s, C = h^s, \{C_y = g^{q_y(0)}, C'_y = H(attr)^{q_y(0)}\}_{y \in Y})$$

The data owner uploads $(CT,\ I_i)$ to cloud storage.

5.3.4 Access Outsourced Data in Public Cloud

To access the EHR of disease d_i, a retriever of the public domain, U_{pub}, sends a request to the server S_{pub}. First, U_{pub} gains a pseudonym list of data owners from S_{pub}. The S_{pub} sends the list of pseudonyms $\{P_o : P_o \in CT\}$ of data owners according to the ciphertexts. When the data retriever U_{pub} decides to retrieve an outsourced data of a data owner with pseudonym P_o, he computes the query $q_i = e(P_o, H(d_i))$. The retriever U_{pub} sends his attribute set $\mathcal{S}$ and q_i to S_{pub}. Next, S_{pub} compares q_i with the values of I_i's received from the data owners. Then, S_{pub} checks whether the attribute set $\mathcal{S}$ satisfies the access policy and sends the authorized ciphertext CT to U_{pub}, upon successful verification. These steps can be described as follows.

(1) $U_{pub} \rightarrow S_{pub}$: $\mathcal{S}$, q_i
(2) $S_{pub} \rightarrow U_{pub}$: CT

It should be noted that a user of the public cloud has only read permission to the outsourced data.

5.3.5 Access Outsourced Data in Private Cloud

The following steps are executed between a retriever of the private domain, U_{pri}, and S_{pri}. Similar to procedure in public domain, the retriever U_{pri} receives a pseudonym list $\{P_o :\ P_o \in CT\}$ of data owners from S_{pri}. Then, he computes the query $q_j = e(P_o, H(d_j))$ to retrieve the EHR of disease d_j for access acc, where acc denotes the request of read or write permission to the outsourced data.

(1) $U_{pri} \rightarrow S_{pri} : \mathcal{S}, q_j, acc$
(2) $S_{pri} \rightarrow U_{pri} : CT$

S_{pri} searches cloud storage according to q_j and the values of indicators I_i's received from the data owners. If the attribute set $\mathcal{S}$ satisfies the access tree of the corresponding ciphertext, then the retriever U_{pri} can access acc to it.

5.3.6 Decryption

To decrypt an EHR, the retriever sends the ciphertext CT and values $\{D_j, D'_j\}$ to the proxy server. The proxy server runs the decryption algorithm. The decryption algorithm takes a ciphertext $CT = (\mathcal{T},\ \tilde{C},\ C,\ \{C_y, C'_y\}_{y \in Y})$ to decrypt it with the private key SK for a leaf node x from the access tree $\mathcal{T}$. If the node x is a leaf node, then sets $i = attr_x$ and computes the following value.

$$\frac{e(D_i,\ C_x)}{e(D'_i,\ C'_x)} = \frac{e(g^r.H(i)^{r_i}, g^{q_x(0)})}{e(g^{r_i},\ H(i)^{q_x(0)})} = e(g,g)^{rq_x(0)}$$

If node x is not a leaf node, the decryption algorithm is called for all children z of x and its output, F_z, will be stored. Let S_x be a k_x-sized set of children nodes of z. The subtree rooted at x is satisfied if and only if k_x subtrees that are rooted at the node x are satisfied. The proxy server computes as follows, where Δ is a Lagrange coefficient.

$$\begin{aligned} F_x,\ where\ & \begin{matrix} S'_x\ index\,(z) : z \in S_x \\ i = index_z \end{matrix} \\ &= \prod_{z \in S_x} (e(g,g)^{r.q_z(0)})^{\Delta_{i,S'_x(0)}} \\ &= \prod_{z \in S_x} (e(g,g)^{r.q_{parent(z)}(index(z))})^{\Delta_{i,S'_x(0)}} \\ &= \prod_{z \in S_x} e(g,g)^{r_i.q_x(i).\Delta_{i,S'_x(0)}} = e(g,g)^{rq_x(0)} \end{aligned}$$

From this recursive algorithm, the proxy server calculates the masking factor essential to decrypt the ciphertext, by calling decryption algorithm with inputs CT, SK and R and acquires $A = e(g,g)^{rq_R(0)} =$

$e(g,g)^{rs}$. Also, the proxy server sends A to the retriever. Then, the retriever restores the original EHR by the following computation.

$$\frac{\tilde{C}}{\{e(C,D)/A\}} = \frac{M\omega^s}{\left\{\frac{e(h^s, g^{(\alpha+r)/\beta})}{e(g,g)^{rs)}}\right\}} = \frac{Me(g,g)^{\alpha s}}{\{e(g^{\beta s}, g^{(\alpha+r)/\beta})/e(g,g)^{rs)}\}} = M$$

5.3.7 Policy Transformation

The policy transformation allows an access control policy to be transformed from a private cloud to a public cloud. When a sensitive record must be accessed by different parties, it is essential to preserve data privacy. For example, consider a researcher who needs a record that can be accessed by the users of the private domain. The policy transformation allows the researcher to read-only access to the desired record while he cannot delete or update it. The policy transformation is performed as follows.

(1) $S_{pub} \rightarrow S_{pri}$: q_i
(2) $S_{pri} \rightarrow U$: CT

The user U requests from S_{pub} to send data about disease d by sending $q = e(P_o, H(d))$ while its ciphertext is in the private cloud. Thus, S_{pub} forwards request to the private cloud S_{pri} to retrieve the corresponding CT. S_{pri} retrieves CT and only gives read permission to access it; this is due to the requester user is in the public domain. Finally, S_{pri} sends the CT to the user U.

5.4 Illustrative Example

For example, consider a data owner with identity ID_1 and with EHR about disease hepatitis determined by d_{hep}. He first receives the key $A_1 = H(ID_1)^{z_1}$ from TA where a random $z_1 \in \mathbb{Z}_p$. Then, the data owner selects $b_1 \in \mathbb{Z}_p$ and computes the parameter $I_{hep} = e(A_1, H(d_{hep})^{b_1})$ and publishes his pseudonym $P_1 = {A_1}^{b_1}$. To access the EHR of disease d_{hep}, a retriever of the public domain, U_{pub}, sends a request to the server S_{pub}. Then, S_{pub} sends the list of pseudonyms of data owners according to the ciphertexts. When the data retriever U_{pub} decides to retrieve an outsourced data of a data owner with a pseudonym P_1, he computes the query $q_{hep} = e(P_1, H(d_{hep}))$. U_{pub} sends his attribute set $\mathcal{S}$ and q_{hep} to S_{pub}. Next, S_{pub} compares q_{hep} with the values of indicators I_i's received from the data owners. Since $q_{hep} = e(P_1, H(d_{hep})) = e({A_1}^{b_1}, H(d_{hep})) = e(A_1, H(d_{hep})^{b_1}) = I_{hep}$, S_{pub} determines whether the attribute set $\mathcal{S}$ satisfies the access policy of q_{hep} and sends the authorized ciphertext CT to U_{pub}. Finally, U_{pub} decrypts the ciphertext CT according to the decryption algorithm described in Section 5.3.6.

6 Security Analysis

In this section, we discuss how the proposed scheme satisfies the following security goals.

6.1 Data Confidentiality

The confidentiality of the proposed scheme is derived from the properties of the CP-ABE scheme [22]. The proposed scheme provides confidentiality of outsourced EHR against the cloud server, unauthorized retrievers and the proxy server. If a cloud server or a user such as a retriever has not enough attribute set for satisfying the access policy, then he cannot acquire the parameter D for decrypting the ciphertexts. Also, since the data is in encrypted form, an adversary cannot learn any information about the outsourced EHR while he has access to the cloud storage. On the other hand, the random value $r \in \mathbb{Z}_p$ used in the private key generation algorithm is secret, thus, the proxy server cannot find the secret value s in the ciphertext although it acquires the parameter $A = e(g,g)^{rs}$. So, the proxy server cannot obtain the EHR in the decryption phase. Consequently, our proposed scheme guarantees EHR confidentiality while provides access control for the cloud-based systems.

Moreover, for data outsourcing, the data owner with identity ID_o and with EHR about disease specified by d_i, randomly selects $b \in \mathbb{Z}_p$ and computes the parameter $I_i = e(A_o, H(d_i)^b)$ as an indicator such that a cloud server uses it to know whether a record matches to the query of a retriever. In a pairing-based cryptography, a pairing between elements of a cryptographic group to other group with a mapping $e : G_1 \times G_1 \rightarrow G_2$ is used. Mapping e is a one-way function that is easy to compute but hard to invert. So, the cloud server cannot obtain any information about A_o or d_i from I_i. Furthermore, the owner generates pseudonym $P_o = {A_o}^b$ and publishes it. According to the discrete logarithm problem (DLP), the cloud server or the retrievers cannot compute the parameter b.

6.2 User Anonymity

In the proposed scheme, a data owner publishes his pseudonym instead of his identity. So, the public/private cloud server requires to solve an instance of the DLP to obtain the identity of the data owner. Also, a data owner sends the value of $e(A_o, H(d_i)^b)$ to the server and the server cannot obtain the key of the data owner. Because of applying attribute-based

encryption, the identities of retrievers are replaced by the attribute sets which must satisfy the access policy. Hence, retriever anonymity is also preserved from the public/private cloud server. Furthermore, values $\{D_j, D'_j\}$ sent from a retriever to a proxy server contain hash values of the retriever attributes. Thus, the proxy server only obtains the corresponding attributes and the retriever identity is also preserved.

6.3 Collusion Resistance

Here, collusion resistance means that the retrievers who have not the necessary attribute set to satisfy the access tree, cannot retrieve any data by colluding with each other. In the key generation phase, the TA selects different values of the parameter r for different retrievers. Hence malicious retrievers cannot collude to construct the value $e(g,g)^{\alpha s}$ and decrypt the encrypted data. In more details, malicious retrievers must use the parameters $\{C_y, C'_y\}$ and $\{D_j, D'_j\}$ from the ciphertexts and their private keys, respectively, for the attribute λ_j. However, to compute F_R in the decryption algorithm, different values of r from different retrievers cannot be integrated to generate $e(g,g)^{\alpha s}$. Moreover, if the proxy server and the public/private cloud server collude with each other cannot compute $e(C,D)/A$ for final decryption. Since, D is an element of the private key of a retriever and only the retriever and the TA know the value of D.

7 Performance Analysis

7.1 Computation Overhead

Regarding the computation cost, we count the number of computational operations of each entity in our system. Table 1 shows the computation overhead of the proposed scheme and compares it with the Rezaeibagha and Mu's scheme [1] and recent research of Zhang *et al.* [13]. In Table 1, we use exp_1 and exp_2 to denote the modular exponentiation in group $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, e for pairing operation, m_1 and m_2 for multiplication in group $\mathbb{G}_1$ and $\mathbb{G}_2$ and Enc for encryption. Let $|\mathcal{S}|$ denotes the size of the attribute set that a data owner uses in EHR encryption and $|U|$ denotes the number of users. In comparison with the Rezaeibagha and Mu's scheme [1], the attribute-based re-encryption scheme [2], by parameter $n \in \mathbb{Z}_p$, is considered; this scheme is consistent with the policy transformation phase.

In the scheme of Rezaeibagha and Mu [1], an EHR must be encrypted by the public keys of all users and the computational cost of encryption phase is multiplied by the number of users. In our scheme, an ABE scheme is used which makes the data owner no need to store the public keys of the retrievers. In comparison with [1], the proposed scheme requires additional computation to access the outsourced data because their scheme does not support a method to search for a disease-based record; they only provide the encrypted data retrieval based on the identity of an EHR. Also, Zhang *et al.* [13] have focused on preserving the user's privacy and hiding access policy. The computation overhead of their scheme for the decryption phase is more than ours. The scheme of Zhang *et al.* [13] is a CP-ABE scheme with efficient decryption and this scheme does not allow searching on encrypted health records.

It should be mentioned that in our scheme one hash, one exponentiation and two bilinear pairings are computed for data outsourcing and retrieving. These computations are efficient in comparison with many attribute-based encryption schemes for cloud storage such as the recent work [23] that used four exponentiations, one hash and four bilinear pairings for data outsourcing and data retrieval.

7.2 Storage Overhead

We analyze the storage overhead of the proposed scheme and compare it with the Rezaeibagha and Mu [1] and Zhang *et al.* [13] in Table 2. Let $|CT|$ and $|\mathcal{T}|$ denotes the size of outsourced ciphertext of the health record and the corresponding access policy, respectively. Also, we assume that the user U_i owns $|\Lambda_i|$ attributes and $|\mathcal{S}|$ attributes exist in the access policy of the ciphertext.

The key generation phase of our scheme has linear overhead toward the attribute set of a retriever and it is similar to [13], but the storage overhead of scheme [1] for this phase is proportional to parameter n. The size of ciphertext is important in terms of the scalability for data retrieval and like [13] it is reasonable to need one copy with essential attributes for encryption. Our scheme supports retrieving health records corresponding to the desired disease while schemes [1, 13] allow data retrieval over encrypted data by record id. So, the proposed scheme has additional overhead for the data access phase.

8 Conclusion

In Rezaeibagha and Mu's scheme, the hybrid access control for an EHR system uses the public and private clouds. Their scheme is based on RBAC and the data owner must determine an access policy for any outsourced EHR. Their scheme only guarantees EHR confidentiality and provides data retrieval based on a record identity. We have used the ciphertext-policy attribute-based encryption and suggested an improved scheme by retrieval capability based on disease. Also, our scheme by utilizing the properties of attribute-based encryption controls the access of

Table 1. Analysis of computation overhead

Phases	Proposed scheme	Rezaeibagha and Mu's scheme [1]	Zhang *et al.*'s scheme [13]
System setup	$2exp_1 + 1exp_2 + 1e$	$9nexp_1 + 1exp_2 + 1e$	$3exp_1 + 1exp_2 + 1e$
Key generation	$(\|U\|\|\Lambda_i\| + 2\|U\|)\,exp_1 + (\|U\|\|\Lambda_i\|)m_1$	$(2n\,\|U\|\,+1)\,exp_1$	$(5\|U\|+2\|\Lambda_i\|\|U\|)exp_1 + (3\|U + \|U\|\|\Lambda_i\|)m_1$
Encryption	$1m_2 + (2\|\mathcal{S}\|+1)exp_1 + 1exp_2$	$1m_2 + (n+3)exp_1 + 1exp_2$	$1e + (2\|\mathcal{S}\|+2)exp_1 + 1exp_2 + (3\|\mathcal{S}\|+1)m_1 + 1m_2$
Access outsourced data in public cloud	$1e$	–	–
Access outsourced data in private cloud	$1e$	–	–
Decryption	$O(\|\mathcal{S}\|+2)m_2 + (2\|\mathcal{S}\|+1)e$	$(n+2)m_2 + (n+1)e$	$O(2\|\mathcal{S}\|m_2) + O(2\|\mathcal{S}\|exp_2) + 2e$
Policy transformation	$1Enc$	$(2n + 1)exp_1 + (n + 1)m_2 + nm_1 + ne$	–

Table 2. Analysis of storage overhead

Phases	Proposed scheme	Rezaeibagha and Mu's scheme [1]	Zhang *et al.*'s scheme [13]
System setup	$3\|\mathbb{G}_1\|+1\|\mathbb{G}_2\|+1\|\mathbb{Z}_p\|$	$(9n+2)\|\mathbb{G}_1\|+1\|\mathbb{G}_2\|$	$5\|\mathbb{G}_1\|+1\|\mathbb{G}_2\|+5\|\mathbb{Z}_p\|$
Key generation	$(2\|\Lambda_i\|+1)\|\mathbb{G}_1\|$	$(2n+1)\|\mathbb{G}_1\|+\|\Lambda_i\|$	$(\|\Lambda_i\|+2)\|\mathbb{G}_1\|$
Ciphertext	$(2\|\mathcal{S}\|+1)\|\mathbb{G}_1\|+\|\mathbb{G}_2\|+\|\mathcal{T}\|$	$1\|\mathbb{G}_2\|+(n+2)\|\mathbb{G}_1\|+\|\mathcal{T}\|$	$(\|\mathcal{S}\|+1)\|\mathbb{G}_1\|+\|\mathbb{G}_2\|+\|\mathcal{T}\|$
Data access	$\|\mathbb{G}_2\|+\|\Lambda_i\|+\|CT\|$	$\|\Lambda_i\|+\|CT\|$	$\|\Lambda_i\|+\|CT\|$

an authorized user to outsource EHRs. Analysis has shown that the proposed scheme effectively achieves data confidentially, user anonymity and collusion resistance. A research direction to explore is designing EHR systems that are operated in a large-scale distributed environment and are managed by multiple authorities who control the access to patient records. In this case, a large amount of data from patients can be hosted on a large-scale distributed system. To provide this possibility, we recommend that the multi-authority ABE [24] can be used in the proposed scheme.

References

[1] Fatemeh Rezaeibagha and Yi Mu. Distributed clinical data sharing via dynamic access-control policy transformation. *International journal of medical informatics*, 89:25–31, 2016.

[2] Xiaohui Liang, Zhenfu Cao, Huang Lin, and Jun Shao. Attribute based proxy re-encryption with delegating capabilities. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pages 276–286, 2009.

[3] Josh Benaloh, Melissa Chase, Eric Horvitz, and Kristin Lauter. Patient controlled encryption: ensuring privacy of electronic medical records. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 103–114, 2009.

[4] Shivaramakrishnan Narayan, Martin Gagné, and Reihaneh Safavi-Naini. Privacy preserving ehr system using attribute-based infrastructure. In *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, pages 47–52, 2010.

[5] Suhair Alshehri, Stanislaw P Radziszowski, and Rajendra K Raj. Secure access for healthcare data in the cloud using ciphertext-policy attribute-based encryption. In *2012 IEEE 28th international conference on data engineering workshops*, pages 143–146. IEEE, 2012.

[6] Changji Wang, Xuan Liu, and Wentao Li. Implementing a personal health record cloud platform using ciphertext-policy attribute-based encryption. In *2012 Fourth International Conference on Intelligent Networking and Collaborative Systems*, pages 8–14. IEEE, 2012.

[7] Xuhui Liu, Qin Liu, Tao Peng, and Jie Wu. Dynamic access policy in cloud-based personal health record (phr) systems. *Information Sciences*, 379:62–81, 2017.

[8] Yan Zhu, Hongxin Hu, Gail-Joon Ahn, Mengyang Yu, and Hongjia Zhao. Comparison-based encryption for fine-grained access control in clouds. In *Proceedings of the second ACM*

conference on Data and Application Security and Privacy, pages 105–116, 2012.

[9] Zhaoquan Cai, Hongyang Yan, Ping Li, Zhengan Huang, and Chongzhi Gao. Towards secure and flexible ehr sharing in mobile health cloud under static assumptions. *Cluster Computing*, 20(3):2415–2422, 2017.

[10] Wei Li, Bonnie M Liu, Dongxi Liu, Ren Ping Liu, Peishun Wang, Shoushan Luo, and Wei Ni. Unified fine-grained access control for personal health records in cloud computing. *IEEE journal of biomedical and health informatics*, 23(3):1278–1289, 2018.

[11] Maithilee Joshi, Karuna Joshi, and Tim Finin. Attribute based encryption for secure access to cloud based ehr systems. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 932–935. IEEE, 2018.

[12] Xiaoling Tao, Chao Lin, Qinglun Zhou, Yong Wang, Kaitai Liang, and Yang Li. Secure and efficient access of personal health record: a group-oriented ciphertext-policy attribute-based encryption. *Journal of the Chinese Institute of Engineers*, 42(1):80–86, 2019.

[13] Leyou Zhang, Gongcheng Hu, Yi Mu, and Fatemeh Rezaeibagha. Hidden ciphertext policy attribute-based encryption with fast decryption for personal health record system. *IEEE Access*, 7:33202–33213, 2019.

[14] Jie Huang, Mohamed Sharaf, and Chin-Tser Huang. A hierarchical framework for secure and scalable ehr sharing and access control in multi-cloud. In *2012 41st International Conference on Parallel Processing Workshops*, pages 279–287. IEEE, 2012.

[15] Ming Li, Shucheng Yu, Kui Ren, and Wenjing Lou. Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings. In *International conference on security and privacy in communication systems*, pages 89–106. Springer, 2010.

[16] Ming Li, Shucheng Yu, Yao Zheng, Kui Ren, and Wenjing Lou. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *IEEE transactions on parallel and distributed systems*, 24(1):131–143, 2012.

[17] Tatiana Ermakova and Benjamin Fabian. Secret sharing for health data in multi-provider clouds. In *2013 IEEE 15th Conference on Business Informatics*, pages 93–100. IEEE, 2013.

[18] Jianghua Liu, Xinyi Huang, and Joseph K Liu. Secure sharing of personal health records in cloud computing: ciphertext-policy attribute-based signcryption. *Future Generation Computer Systems*, 52:67–76, 2015.

[19] Ye Li, Kaitai Liang, Chunhua Su, and Wei Wu. Dabehr: decentralized attribute-based electronic health record system with constant-size storage complexity. In *International Conference on Green, Pervasive, and Cloud Computing*, pages 611–626. Springer, 2017.

[20] R Charanya, S Nithya, and N Manikandan. Attribute based encryption for secure sharing of e-health data. In *Materials Science and Engineering Conference Series*, volume 263, page 042030, 2017.

[21] Gandikota Ramu, B Eswara Reddy, Appawala Jayanthi, and LV Narasimha Prasad. Fine-grained access control of ehrs in cloud using cp-abe with user revocation. *Health and Technology*, 9(4):487–496, 2019.

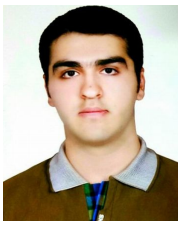
[22] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE symposium on security and privacy (SP'07)*, pages 321–334. IEEE, 2007.

[23] Shangping Wang, Shasha Jia, and Yaling Zhang. Verifiable and multi-keyword searchable attribute-based encryption scheme for cloud storage. *IEEE Access*, 7:50136–50147, 2019.

[24] Allison Lewko and Brent Waters. Decentralizing attribute-based encryption. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 568–588. Springer, 2011.

Maryam Zarezadeh received her B.Sc. degree in Information Technology (IT) Engineering from University of Isfahan in 2010 and M.Sc. degree in IT Engineering (Information Security) from Shahed University in 2013. She is currently a Ph.D. student in IT Engineering (Information Security) at University of Isfahan. Her research interests are security protocols and network security.

Maede Ashouri-Talouki is an Assistant Professor of IT Engineering department of University of Isfahan (UI). She received her B.S., M.S., and Ph.D. degrees from University of Isfahan in 2004, 2007 and 2012, respectively. In 2013, she joined University of Isfahan. Her research interests include mobile networks security, user privacy and anonymity, cryptographic protocols and network security.



Mohammad Siavashi received his B.Sc degree in computer engineering from Shiraz University, Iran, in 2019. He developed a solid experience in software engineering, blockchain, and cybersecurity. He co-founded two blockchain startups in silicon valley during the past two years. His research interests include but not limited to cybersecurity, software engineering, machine learning, and blockchain.