UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

UPCommons

Portal del coneixement obert de la UPC

http://upcommons.upc.edu/e-prints

Aquesta és una còpia de la versió *author's final draft* d'un article publicat a la revista *Journal of supercomputing*.

La publicació final està disponible a Springer a través de http://dx.doi.org/ 10.1007/s11227-014-1253-3

This is a copy of the author 's final draft version of an article published in the journal *Journal of supercomputing*.

The final publication is available at Springer via http://dx.doi.org/10.1007/s11227-014-1253-3

Article publicat / Published article:

Xhafa, F. [et al.] (2015) Privacy-aware attribute-based PHR sharing with user accountability in cloud computing. "Journal of supercomputing". Vol. 71, Issue. 5. p.1607-1619. Doi: 10.1007/s11227-014-1253-3

Noname manuscript No.
(will be inserted by the editor)

Privacy-Aware Attribute-Based PHR Sharing with User Accountability in Cloud Computing

Fatos Xhafa · Jianglang Feng* · Yinghui Zhang · Xiaofeng Chen · Jin Li

Received: date / Accepted: date

Abstract As an emerging patient-centric model of health information exchange, personal health record (PHR) is often outsourced to be stored at a third party. The value of PHR data is its long-term cumulative record relevant with personal health which can be significant in the future when faced with disease occurrences. As a promising public key primitive, attribute-based encryption (ABE) has been used to design PHR sharing systems. However, the existing solutions fail to achieve several important security objectives, that is, no need for a single authority to issue private keys to all PHR users, user access privacy protection, and user accountability. In this paper, we propose

Fatos Xhafa
Department de Llenguatges i Sistemes Informatics, Universitat Politćcnica de Cataunya Barcelona, Spain
E-mail: fatos@lsi.upc.edu

Jianglang Feng
Department of Mathematics, College of Management Science, Chengdu University of Technology, Chengdu, P.R. China
* Corresponding author
E-mail: 43601361@qq.com

Yinghui Zhang
National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an, P.R. China;
and State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, P.R. China
E-mail: yhzhaang@163.com

Xiaofeng Chen
State Key Laboratory of Integrated Service Networks (ISN), Xidian University, Xi'an, P.R. China
E-mail: xfchen@xidian.edu.cn

Jin Li
School of Computer Science and Educational Software, Guangzhou University, Guangzhou, P.R. China
E-mail: jinli71@gmail.com

a multi-authority ciphertext-policy ABE scheme with user accountability and apply it to design an attribute-based PHR sharing system. In the proposed solution, the access policy is hidden and hence user access privacy is protected. In particular, the global identity of a misbehaving PHR user who leaked the decryption key to other unauthorized users can be traced, and thus the trust assumptions on both the authorities and the PHR users are reduced. Extensive analysis shows that the proposed scheme is provably secure and efficient.

Keywords Personal health record · Cloud computing · Multi-authority · Anonymity · Accountability

1 Introduction

In recent times, there has been a remarkable upsurge in activity surrounding the utilization of personal health record (PHR) systems for individuals such as patients and consumers. The value of PHR data is its long-term cumulative record relevant with personal health which can be significant in the future when faced with disease occurrences. According to the definition in [1], the PHR is an electronic application through which people can access and coordinate their health information and share parts of it with those who need it in a private and secure environment. In part, PHR systems can be seen as a repository for patient health data. Besides providing the fundamental repository service to individuals to store their PHR data, PHR systems also include other desirable functionality such as the functionality of decision support. They can help patients make the best decision to improve their health care quality.

With the advent of cloud computing, healthcare service providers have moved their PHR data to public clouds. Google and Microsoft are two primary cloud service providers, and they can offer PHR services based on their cloud platforms. To eliminate the risk of privacy exposure, PHR service providers should not only encrypt patients' data, but also allow PHR owners to control with whom they intend to share PHR data. That is, PHR systems should realize patient-centric model of health information exchange. To assure the patients' full control over their own PHR data, there has been an increasing interest in applying attribute-based encryption (ABE) to realize secure PHR sharing. Ibraimi et al. [2] applied ciphertext policy ABE (CP-ABE) [3] to manage the sharing of PHR data. Akinyele et al. [4] adopted ABE to generate self-protecting electronic healthcare records (EMRs), which can either be stored on cell phones so that EMR could be accessed even if the health provider is offline. However, the above solutions usually use a single attribute authority (AA), which is trusted by all users in the system. In addition, it is not practical to delegate all attribute management tasks to one AA, including certifying all users' attributes and generating attribute private keys. Recently, Li et al. [12, 5] proposed a novel patient-centric framework and a suite of mechanisms for data access control to PHR data stored in semi-trusted cloud servers. Similar to the technique in [2], the PHR users in the sharing system in [5] are divided into multiple security domains, which greatly reduces the complexity of key management for PHR owners and PHR users. To guarantee patient privacy, multi-authority ABE [6] are exploited in [5,13]. Lu et al. [7] proposed a secure and privacy-preserving opportunistic computing framework for mobile healthcare emergency. There are also many other healthcare-related solutions [8,9,10] proposed to realize secure PHR data sharing.

Although the above schemes apply ABE to design PHR sharing systems, there is an important security aspect, user accountability, has not be formally addressed. This problem is extremely important in that attribute private keys directly imply PHR users' privileges to the protected data in the attribute-based setting. The dishonest PHR users may share their attribute private keys with unauthorized users. They can just directly give part of their original or transformed private keys such that nobody can tell who has disclosed these keys. This will violate the privacy protection of patients. To our knowledge, the issue of user accountability in PHR sharing systems based on multi-authority ABE is quite new and has not been solved yet.

Our Contribution. In order to realize user accountability in PHR sharing systems, in this paper, we propose a multi-authority CP-ABE scheme with user accountability. The supported policy is AND gates on multiple attribute values and wildcards. In our scheme, a PHR user obtains his attribute private key and if the attribute set associated with the private key does not satisfy the access policy in a PHR ciphertext, the PHR user cannot decrypt and guess what access policy was specified by the PHR owner. Hence, the access policy is hidden and user access privacy is protected. We apply the proposed scheme to design an attribute-based PHR sharing system, which allows to trace the global identity of a misbehaving PHR user who leaked the decryption key to others, and thus reduces the trust assumptions not only on the authorities but also the PHR users. Extensive analysis shows that the proposed scheme is secure and efficient.

1.1 Related Work

Since the introduction of ABE in implementing fine-grained access control systems [11], a lot of works have been proposed to design flexible ABE schemes. ABE comes in two flavors called key-policy ABE (KP-ABE) and CP-ABE [14]. The first KP-ABE construction [14] realized the monotonic access policy for key policies. Bethencourt et al. proposed the first CP-ABE construction [3].

To further achieve users' attribute privacy protection, anonymous ABE [15] has received a lot of attention. For practicality, a more efficient anonymous CP-ABE scheme was constructed [16]. In particular, Zhang et al. [17] introduced a novel technique called match-then-decrypt into the decryption of anonymous ABE, which can greatly improve the decryption efficiency of anonymous ABE. Although anonymous ABE can realize secure anonymous PHR sharing, before its widely deployment, another important security issue, user accountability, has to be addressed. Several attempts [18,19,20] have been made to address the accountability problem in attribute-based access control.

In [18], they considered how to defend the key-abuse problem in KP-ABE. In addition to the user accountability, the user attribute privacy is also taken into consideration in [19]. However, these schemes may not be entirely realistic, in that they assume the existence of a single trusted party who monitors all attributes and generates all decryption keys. In order to reduce the trust assumption, Li et al. [21] proposed a multi-authority CP-ABE scheme with user accountability, where each attribute has two values. To our knowledge, there are no multi-authority attribute-based PHR sharing systems, which simultaneously support users' attribute privacy protection and accountability.

2 Preliminaries

2.1 Notations

Let $[N] = \{1, 2, \cdots, N\}$ and U be the universal attribute set of the PHR sharing system. We denote by U_k the attribute set managed by the k-th attribute authority AA_k. Let $\omega_{k,i} \in S_{k,i}$ be the i-th attribute distributed by AA_k.

2.2 Bilinear Pairings

Let $\mathbb{G}$ and $\mathbb{G}_T$ be cyclic multiplicative groups of some large prime order p and we denote the identity of $\mathbb{G}_T$ as 1. We call $\hat{e}$ a bilinear pairing if $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a map with the following properties:

1. Bilinear: $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$ for all $a, b \in_R \mathbb{Z}_p$.
2. Non-degenerate: There exists $g_1, g_2 \in \mathbb{G}$ such that $\hat{e}(g_1, g_2) \neq 1$.
3. Computable: It is feasible to compute $\hat{e}(g_1, g_2)$ for all $g_1, g_2 \in_R \mathbb{G}$.

2.3 Access Structure

As a generalization of access structures in [22], the adopted access structures are the same as those in [16,19]. L satisfies W is represented as $L \models W$, and the case of L does not satisfy W is denoted by $L \not\models W$. Formally, given an attribute list $L = [L_1, L_2, \cdots, L_n]$ and an access structure $W = [W_1, W_2, \cdots, W_n]$, $L \models W$ if $L_i = W_i$ or $W_i = *$ for all $1 \leq i \leq n$, and otherwise $L \not\models W$. It is noted that the wildcard $*$ in W plays the role of "don't care" value.

3 System Model and Design Goals

3.1 System Architecture

As shown in Fig. 1, the architecture of the accountable attribute-based PHR sharing system in cloud storage consists of five entities. We only describe

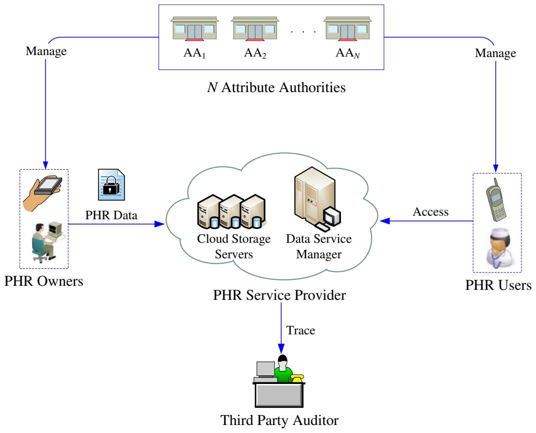

Fig. 1 Architecture of attribute-based PHR sharing system in cloud

the third party auditor (TPA). The other enties are the same as those in typical ABE system. The TPA is a third parity auditor, who can trace the identity of dishonest users. These dishonest users may share their attribute private keys with other users, who do not have these privileges. In our system, user accountability can be achieved in the black-box model by embedding additional user-specific information into the attribute private key, while still hiding access policies.

3.2 Security Goals

To realize "patient-centric" PHR sharing, a core requirement is that each patient can easily control who can access his own PHR data. The potential threats of the system and our security goals are summarized as follows:

- **Confidentiality.** By confidentiality we mean that unauthorized PHR users (may be the PHR service provider and adversaries) who do not have enough attributes matching the access policy specified for a ciphertext by a PHR owner should be prevented from accessing the PHR data.
- **Collusion-Resistance.** Even if multiple PHR users collude, they cannot access the plaintext of a PHR ciphertext if each user cannot decrypt the ciphertext alone.
- **Key-Escrow Freeness.** In most of the existing PHR sharing systems, a central attribute authority is introduced, which requires too much trust on a single authority. Furthermore, a single authority may be a performance bottleneck of the system when there are a large number of PHR users.
- **Attribute Privacy Protection.** In PHR sharing systems, the access policy itself could be sensitive information and needs to be protected.
- **Accountability.** In attribute-based PHR sharing systems, the attribute private key directly imply users' privileges to the protected PHR data. The problem of key abuse is important and should be prevented.

3.3 Definition of Multi-Authority CP-ABE with Accountability

A N-authority CP-ABE scheme with accountability consists of the following five algorithms:

- $\mathsf{Setup}(1^\lambda,\ N) \rightarrow (\mathsf{params}, \{(APK_k, ASK_k)\}_{k\in[N]})$: The randomized setup algorithm takes as input a security parameter $\lambda \in \mathbb{N}$ and the total number of attribute authorities $N \in \mathbb{N}$, and outputs the system parameters params, N public/private key pairs $\{(APK_k, ASK_k)\}_{k\in[N]}$ for the N attribute authorities, respectively. We assume that the other algorithms take params and $\{(APK_k)\}_{k\in[N]}$ as implicit inputs. Also, the attribute domains managed by each attribute authority is included in params.
- $\mathsf{AttKeyGen}(ASK_k, L_k, \mathrm{GID}) \rightarrow SK_{k,L_k,ID}$: The randomized key generation algorithm is run by the k-th attribute authority AA_k. On input the its private key ASK_k, an attribute list $L_k = \{L_{k,1}, L_{k,2}, \cdots, L_{k,n_k}\}$ and a global identity GID, it outputs $SK_{k,L_k,ID}$ as a decryption key corresponding to the attribute list L_k for the PHR user with the global identity GID.
- $\mathsf{Encrypt}(M, W) \rightarrow CT_W$: The randomized encryption algorithm is run by the encryptor. On input a message M and a ciphertext policy $W = [W_1, W_2, \cdots, W_N]$, it generates a ciphertext CT_W as the encryption of M with respect to W, where W_k reflects a subset of the attribute domain managed by the k-th attribute authority.
- $\mathsf{Decrypt}(CT_W, SK_{L,ID}) \rightarrow M$ or $\perp$: The decryption is run by the PHR users. On input a ciphertext CT_W of a message M under a ciphertext policy W, and a secret key $SK_{L,ID} = \{SK_{k,L_k,ID}\}_{k\in[N]}$ associated with L and GID, the ciphertext CT_W is successfully decrypted to recover the message M if $(L \parallel \mathrm{GID}) \models W$. Otherwise, the algorithm returns $\perp$.
- $\mathsf{Trace}^{\mathcal{D}}(W)$: This is an oracle algorithm for recovering the global identity GID related to the decryption private key incorporated in a private device $\mathcal{D}$. It takes as inputs the public parameters and the ciphertext policy W, and outputs a global identity.

4 Privacy-Aware Attribute-Based PHR Sharing System with User Accountability

4.1 Overview of the Proposed Solution

In our solution, a PHR user is given an attribute private key associated with $L \parallel \mathrm{GID}$, where L represents an attribute list and GID is the PHR user's global identity. In order to realize attribute privacy protection, for each attribute value $v_{k,i,t}$, we compute four ciphertext components $C_{k,i,t,0}$, $C_{k,i,t,1}$,$\widehat{C}_{k,i,t,0}$,$\widehat{C}_{k,i,t,1}$. Based on the components specified by indexes of the attribute secret key, a recipient can decrypt a ciphertext without knowing the potential ciphertext policy. To make it hard to distinguish the well-formed ciphertext component from the malformed one, we use the linear splitting technique [23]. The ciphertext is computed by splitting the random value s into two parts $s_{k,i,t}$ and $s - s_{k,i,t}$. To achieve user accountability, in normal encryption, a PHR data is encrypted under a ciphertext policy $W = W' \parallel *$ such that any PHR user with $L \parallel \mathrm{GID}$ satisfying $(L \parallel \mathrm{GID}) \models W$ is able to decrypt, regardless of the user's identity. Obviously, this holds because the second part in the W is "don't care". In the tracing algorithm, a PHR data is encrypted with $W = W' \parallel \mathrm{GID}^*$ to test whether the identity in the private device is GID. It follows from the anonymity that the ciphertext is indistinguishable from other ciphertexts under the ciphertext policy $W' \parallel *$. And, only users with attribute private key associated with $L \parallel \mathrm{GID}$ satisfying $(L \parallel \mathrm{GID}) \models W$ can decrypt the ciphertext. As a result, the global identity GID^* can be determined in the private device. Finally, in order to eliminate central authorities, we use a set of pseudorandom functions.

4.2 Description of the Proposed PHR Sharing System

- **System Initialization:** Assume that the universe of attributes in the PHR sharing system is denoted by $U = \{\omega_{k,i}\}_{k \in [N], i \in n_k}$. Then, based on a security parameter 1^λ and the following system setup algorithm Setup, the public system parameter params and N public/private key pairs for the N attribute authorities are generated, respectively.
  Setup: Let $\mathbb{G}$, $\mathbb{G}_T$ be two cyclic multiplicative groups of prime order p, and $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear pairing. Let g_1, g_2 be random elements from $\mathbb{G}$. Define two hash functions $H_0 : \{0,1\}^\rho \rightarrow \mathbb{G}$ and $H : \{0,1\}^* \rightarrow \mathbb{G}$. Let $\mathrm{AA}_1, \mathrm{AA}_2, \cdots, \mathrm{AA}_N, \mathrm{AA}_{N+1}$ be the $N+1$ attribute authorities in the system. Each authority AA_k is in charge of a disjoint set of n_k attributes. Let the value set of the i-th attribute managed by authority AA_k be $\mathbb{S}_k = \{\omega_{k,i} | 1 \leq i \leq n_k\}$, and $S_{k,i} = \{v_{k,i,t} | 1 \leq t \leq m_{k,i}\}$ be the value set of the attribute $\omega_{k,i}$, where $n_k, m_{k,i} \in \mathbb{N}$. Also, the set of attributes managed by authority AA_{N+1} is the set of users' global identities, i.e., $\omega_{N+1,i} \in \{0,1\}$ for all $1 \leq i \leq n_{N+1} = \rho$, the bit-length of an identity with $2^\rho < p$. Then, the following steps are fulfilled:
  1. For $1 \leq k \leq N+1$, the attribute authority AA_k chooses $x_k \in_R \mathbb{Z}_p^*$ as his secret key, computes $y_k = g_1^{x_k}$ and sends $\hat{e}(g_1, g_2)^{x_k}$ to the other attribute authorities.
  2. Every attribute authority can compute a system public key Y as
  $$Y = \hat{e}(\sum_{k=1}^{N+1} y_k, g_2) = \sum_{k=1}^{N+1} \hat{e}(g_1, g_2)^{x_k}.$$

  Finally, the system public parameters are $\mathsf{params} = (g_1, g_2, Y, H_0, H)$.
- **PHR User Grant**: Assuming a new PHR user with global identity GID is intended to join the PHR sharing system, he needs to request an attribute private key associated with his attribute list L from the attribute authorities. Specifically, the attribute authorities run the following key generation algorithm $\mathsf{AttKeyGen}$ to generate the private key for the PHR user.
  AttKeyGen: The attribute private key is generated in the following:

1. Each attribute authority AA_k shares a secret pseudorandom function (PRF) seed $s_{k,k'} \in \mathbb{Z}_p^*$ with $\mathrm{AA}_{k'}$. It also chooses a PRF seed $a_k \in \mathbb{Z}_p^*$ and computes $y'_k = g_1^{a_k}$, which is sent to all the other attribute authorities. It then defines a pseudorandom function $\mathrm{PRF}_{k,k'}(\mathrm{GID}) = g_1^{\frac{a_k a'_k}{s_{k,k'}+X}}$, where $X = H_0(\mathrm{GID})$.
2. To generate an attribute private key for the attribute list $L = L_1 \parallel L_2 \parallel \cdots \parallel L_N$, the user with global identity $\mathrm{GID} = (I_1, I_2, \cdots, I_\rho) \in \{0,1\}^\rho$ first gets $\{D_{k,j}\}$ for $k \neq j$ by using the anonymous key-issuing protocol with the k-th authority. In more details, the user starts N indpendent invocations of the anonymous key-issuing protocol on input $(y_j^{a_k}, g_1, \delta_{k,j}R_{k,j}, s_{k,j}, \delta_{k,j})$ with the k-th authority, where $R_{k,j} \in_R \mathbb{Z}_p^*$ is randomly chosen by the authority AA_k, and $\delta_{k,j}$ is 1 if $k > j$ and -1 otherwise, for $j \in [N+1] - \{k\}$. At the end, the user obtains $E_{k,j} = g_1^{R_{k,j}}\mathrm{PRF}_{k,j}(\mathrm{GID})$ if $k > j$, and $E_{k,j} = \frac{g_1^{R_{k,j}}}{\mathrm{PRF}_{k,j}(\mathrm{GID})}$ otherwise.
3. Note that, the PRF values for the same global identity GID from multiple authorities cancel each other. After interacted with all $N+1$ authorities, it follows that $E = \prod_{k,k' \in [N+1], k \neq k'} E_{k,k'} = g_1^R$ where $R = \sum_{k,k' \in [N+1], k \neq k'} R_{k,k'}$.
4. In order to generate an attribute private key for an attribute list $L_k = \{L_{k,1}, L_{k,2}, \cdots, L_{k,n_k}\} = \{v_{k,1,t_{k,1}}, v_{k,2,t_{k,2}}, \cdots, v_{k,n_k,t_{k,n_k}}\} \subseteq \mathbb{S}_k$ from the k-th attribute authority, AA_k picks up $r_{k,1}, r_{k,2}, \cdots, r_{k,n_k-1}$, $\lambda_{k,1}$, $\lambda_{k,2}, \cdots, \lambda_{k,n_k}, \widehat{\lambda}_{k,1}, \widehat{\lambda}_{k,2}, \cdots, \widehat{\lambda}_{k,n_k} \in_R \mathbb{Z}_p^*$ and also computes $r_{k,n_k} = x_k - \sum_{i=1}^{n_k-1} r_{k,i} - \sum_{k' \in [N+1]-k} R_{k,k'}$. Then, the private key component is computed as $SK_{k,L_k,ID} = \{D_{k,i,0}, D_{k,i,1}, \widehat{D}_{k,i,0}, \widehat{D}_{k,i,1}\}_{i \in [n_k]} = \{g_2^{\lambda_{k,i}}, g_1^{r_{k,i}} H(0||k||i||v_{k,i,t_{k,i}})^{\lambda_{k,i}}, g_1^{\widehat{\lambda}_{k,i}}, g_2^{r_{k,i}} H(1||k||i||v_{k,i,t_{k,i}})^{\widehat{\lambda}_{k,i}}\}_{i \in [n_k]}$.
5. AA_{N+1} chooses $\{r_i \in \mathbb{Z}_p^*\}_{i \in [\rho-1]}$, $\{\lambda_i, \widehat{\lambda}_i \in \mathbb{Z}_p^*\}_{i \in [\rho]}$ and computes $r_\rho = x_{N+1} - \sum_{i=1}^{\rho-1} r_i - \sum_{k' \in [N]} R_{N+1,k'}$. Then $SK_{ID} = \{D_{j,0}, D_{j,1}, \widehat{D}_{j,0}, \widehat{D}_{j,1}\}_{j \in [\rho]} = \{g_2^{\lambda_j}, g_1^{r_j} H(0||N+1||j||I_j)^{\lambda_j}, g_1^{\widehat{\lambda}_j}, g_2^{r_j} H(1||N+1||j||I_j)^{\widehat{\lambda}_j}\}_{j \in [\rho]}$.
6. Finally, $SK_{L,ID} = \{E, \{SK_{k,L_k,ID}\}_{k \in [N]}, SK_{ID}\}$.

- **PHR File Storage**: During this part, the PHR owner encrypts his private PHR file F and uploads the resulted ciphertexts to the cloud storage server managed by the PHR service provider. Whenever the PHR owner intends to create and upload a file F to the cloud servers, he first defines an access policy for the file, which is represented by $W = [W_1, W_2, \cdots, W_N] \wedge *$, where $W_k = \{W_{k,i}\}_{1 \leq i \leq n_k}$ for $k \in [N]$. Each $W_{k,i}$ is chosen from the value set $S_{k,i} \cup \{*\}$ of the attribute $\omega_{k,i}$. For instance, $W_{k,i}$ could be an attribute like "profession=physician" while the attribute "profession" has multiple values. Note that, the PHR owner just sets $W_{k,i} = *$ to indicate that the owner does not care the attribute $\omega_{k,i}$ in the access policy. Subsequently, the PHR owner randomly picks a symmetric key K from the key space and encrypts the file F based on K using a standard symmetric key encryption algorithm such as AES to generate a ciphertext C_F. Then, he runs the following attribute-based encryption algorithm $\mathsf{Encrypt}$ on (M, W) and obtains the ciphertext $C_M \triangleq CT_W$ of the symmetric key K with respect to the access policy W, where $M = TH(K)$. It is noted that TH is a trapdoor hash function with a trapdoor TD such that $M \in \mathbb{G}_T$. Finally, the PHR owner uploads (C_F, TD, C_M) to the cloud storage server.

Encrypt: The attribute-based encryption algorithm proceeds as follows: To encrypt $M \in_R G_T$ under the access policy $W = [W_1, W_2, \cdots, W_N]$, the PHR owner chooses $s \in_R \mathbb{Z}_p^*$, and computes $C_0 = MY^s$ and $\widehat{C} = g_2^s$. Then the following steps are performed.

1. For each $1 \leq k \leq N$, the PHR owner parses $W_k = \{W_{k,i}\}_{1 \leq i \leq n_k}$ and for $1 \leq i \leq n_k$, the following two circumstances are considered:
   - **Case 1:** If $v_{k,i,t} \in W_{k,i}$, the PHR owner chooses $s_{k,i,t} \in_R \mathbb{Z}_p^*$ and computes
   $$\begin{aligned}&\left\{C_{k,i,t,0}, C_{k,i,t,1}, \widehat{C}_{k,i,t,0}, \widehat{C}_{k,i,t,1}\right\}\\ =& \left\{H(0||k||i||v_{k,i,t})^{s_{k,i,t}}, g_2^{s_{k,i,t}}, H(1||k||i||v_{k,i,t})^{s-s_{k,i,t}}, g_1^{s-s_{k,i,t}}\right\}.\end{aligned}$$
   - **Case 2:** If $v_{k,i,t} \notin W_{k,i}$, the PHR owner chooses $s_{k,i,t}, s'_{k,i,t} \in_R \mathbb{Z}_p^*$ and computes
   $$\begin{aligned}&\left\{C_{k,i,t,0}, C_{k,i,t,1}, \widehat{C}_{k,i,t,0}, \widehat{C}_{k,i,t,1}\right\}\\ =& \left\{H(0||k||i||v_{k,i,t})^{s_{k,i,t}}, g_2^{s_{k,i,t}}, H(1||k||i||v_{k,i,t})^{s'_{k,i,t}}, g_1^{s'_{k,i,t}}\right\}.\end{aligned}$$
2. For $k = N + 1$, the PHR owner chooses $s_j, s'_j \in_R \mathbb{Z}_p^*$, then for each $1 \leq j \leq \rho$, computes
   $$\{C_{0,j,0}, C_{0,j,1}, \widehat{C}_{0,j,0}, \widehat{C}_{0,j,1}\} = \{H(0||N{+}1||j||0)^{s'_j}, g_2^{s'_j}, H(1||N{+}1||j||0)^{s-s'_j}, g_1^{s-s'_j}\},$$
   and
   $$\{C_{1,j,0}, C_{1,j,1}, \widehat{C}_{1,j,0}, \widehat{C}_{1,j,1}\} = \{H(0||N{+}1||j||1)^{s_j}, g_2^{s_j}, H(1||N{+}1||j||1)^{s-s_j}, g_1^{s-s_j}\}.$$

Finally, the attribute-based ciphertext is

$$\begin{aligned}CT_W = \Big\{&C_0, \widehat{C}, \{C_{k,i,t,0}, C_{k,i,t,1}, \widehat{C}_{k,i,t,0}, \widehat{C}_{k,i,t,1}\}_{k\in[N], i\in[n_k], t\in[m_{k,i}]},\\ &\{C_{i,j,0}, C_{i,j,1}, \widehat{C}_{i,j,0}, \widehat{C}_{i,j,1}\}_{i\in\{0,1\}, j\in[\rho]}\Big\}.\end{aligned}$$

- **PHR File Access:** When a PHR user wants to access the PHR file F, he sends the request message to the PHR service provider. The PHR service provider sends back the ciphertext (C_F, TD, C_M) to the user. If the PHR user has the privilege to access the PHR file F, he can decrypt to get the trapdoor hash value $M = TH(K)$ of the symmetric key K through the following attribute-based decryption algorithm $\mathsf{Decrypt}$ on $CT_W = C_M$. Then, the user computes the symmetric key K based on M and the trapdoor TD. Finally, the PHR file can be obtained by performing the symmetric decryption algorithm based on C_F and K.

Decrypt: Suppose a user with $SK_{L,ID} = \{E, \{SK_{k,L_k,ID}\}_{k\in[N]}, SK_{ID}\}$ for an attribute list $L = \{L_k\}_{k\in[N]}$ and GID $= (I_1, I_2, \cdots, I_\rho) \in_R \{0,1\}^\rho$ wants to decrypt the ciphertext CT_W, where $L_k = \{L_{k,1}, L_{k,2}, \cdots, L_{k,n_k}\} = \{v_{k,1,t_{k,1}}, v_{k,2,t_{k,2}}, \cdots, v_{k,n_k,t_{k,n_k}}\}$. Then

1. To decrypt the ciphertext without knowing the ciphertext policy W, the PHR user computes C_1 and C_2 in the following:
   - Suppose the indexes satisfy $L_{k,i} = v_{k,i,t}$, C_1 is computed as
   $$C_1 = \frac{\prod_{k=1}^{N}\prod_{i=1}^{n_k} \hat{e}(C_{k,i,t,0}, D_{k,i,0})\hat{e}(\widehat{C}_{k,i,t,0}, \widehat{D}_{k,i,0})}{\prod_{k=1}^{N}\prod_{i=1}^{n_k} \hat{e}(C_{k,i,t,1}, D_{k,i,1})\hat{e}(\widehat{C}_{k,i,t,1}, \widehat{D}_{k,i,1})}. \tag{1}$$
   - C_2 is computed by taking the following two cases into account:
   $$C_2 = \frac{\prod_{i=1}^{\rho} \hat{e}(C_{I_j,j,0}, D_{j,0})\hat{e}(\widehat{C}_{I_j,j,0}, \widehat{D}_{j,0})}{\prod_{i=1}^{\rho} \hat{e}(C_{I_j,j,1}, D_{j,1})\hat{e}(\widehat{C}_{I_j,j,1}, \widehat{D}_{j,1})}. \tag{2}$$
2. Finally, the PHR user can achieved $M = C_0(C_1C_2\hat{e}(E, \widehat{C}))^{-1}$.

- **Black-Box Tracing:** Suppose an illegal device is found to be used to access the PHR files stored in the could storage servers. In order to pinpoint the identity of the dishonest user, the TPA can adopt the following tracing algorithm Trace.

  Trace: Suppose the ciphertext policy is W. the TPA does:

  1. Extracts $L_0 = \{L_{k_j,i_{j,t}}\}_{1\le j\le\tau, 1\le t\le t_j}$ from W, where $1 \le k_j \le N$, $1 \le t_j \le n_{k_j}$, $1 \le i_{j,t} \le n_{k_j}$ for $1 \le j \le \tau$, $1 \le t \le t_j$. The values in positions except those in Index$_0 = \{\{(k_j, i_{j,t})\}_{1\le j\le\tau,1\le t\le t_j}\}$ are $*$.
  2. For a suspicious user set S, in which the users have the attributes associated with L_0, there are two ways to pinpoint the identity from S.
     Case 1: The size of set S is not huge. In this case, the TPA just encrypts some message with respect to W for each GID $\in S$ until the identity is determined. To encrypt a message $M \in \mathbb{G}_T$ under the access policy $W = [W_1, W_2, \cdots, W_N] \wedge W_{N+1}$, where $W_{N+1} =$ GID, the TPA first chooses $s \in_R \mathbb{Z}_p^*$, computes $C_0 = MY^s$, and then does the following:
     (a) For each $1 \le k \le N$, parse $W_k = \{W_{k,i}\}_{1\le i\le n_k}$. For $1 \le i \le n_k$, it does: If $v_{k,i,t} \in W_{k,i}$, choose $s_{k,i,t} \in_R \mathbb{Z}_p^*$ and compute the components $\{C_{k,i,t,0}, C_{k,i,t,1}, \widehat{C}_{k,i,t,0}, \widehat{C}_{k,i,t,1}\}$ as $\{H(0||k||i||v_{k,i,t})^{s_{k,i,t}}, g_2^{s_{k,i,t}}$, $H(1||k||i||v_{k,i,t})^{s-s_{k,i,t}}, g_1^{s-s_{k,i,t}}\}$. If $v_{k,i,t} \notin W_{k,i}$, choose $s_{k,i,t}, s'_{k,i,t} \in_R$ $\mathbb{Z}_p^*$ and compute the components as $\{H(0||k||i||v_{k,i,t})^{s_{k,i,t}}, g_2^{s_{k,i,t}}$, $H(1||k||i||v_{k,i,t})^{s'_{k,i,t}}, g_1^{s'_{k,i,t}}\}$.
     (b) For $k = N+1$, assume GID $= \{I_1, I_2, \cdots, I_\rho\}$, choose $s_j, s'_j, s''_j \in_R$ $\mathbb{Z}_p^*$, then for each $1 \le j \le \rho$, the following two circumstances are considered: If $I_j = 0$, compute $\{C_{0,j,0}, C_{0,j,1}, \widehat{C}_{0,j,0}, \widehat{C}_{0,j,1}\} =$ $\{H(0||N{+}1||j||0)^{s_j}, g_2^{s_j}, H(1||N{+}1||j||0)^{s-s_j}, g_1^{s-s_j}\}$ and $\{C_{1,j,0}, C_{1,j,1},$ $\widehat{C}_{1,j,0}, \widehat{C}_{1,j,1}\} = \{H(0||N+1||j||1)^{s'_j}, g_2^{s'_j}, H(1||N+1||j||1)^{s''_j}, g_1^{s''_j}\}$. If $I_j = 1$, compute $\{C_{0,j,0}, C_{0,j,1}, \widehat{C}_{0,j,0}, \widehat{C}_{0,j,1}\} = \{H(0||N{+}1||j||0)^{s'_j}$, $g_2^{s'_j}, H(1||N+1||j||0)^{s''_j}, g_1^{s''_j}\}$, and $\{C_{1,j,0}, C_{1,j,1}, \widehat{C}_{1,j,0}, \widehat{C}_{1,j,1}\} =$ $\{H(0||N+1||j||1)^{s_j}, g_2^{s_j}, H(1||N+1||j||1)^{s-s_j}, g_1^{s-s_j}\}$.

It can be easily seen that a PHR user is able to decrypt the ciphertext only when his global identity is GID and he has the attribute list L_0.

Case 2: The size of set S is huge, and the tracing algorithm proceeds:

(a) First, the TPA tries an attribute value $L_{k,i}$ from the position (k, i) where $W_{k,i} = *$. Then, it encrypts a message using the normal encryption algorithm with respect to W' such that all positions are set to be $*$, except the positions defined by $\text{Index}_0 \cup (k, i)$ are set to be $L'_0 = L_0 \cup L_{k,i}$.

(b) The ciphertext is sent to the private device. If the ciphertext can be decrypted correctly, the TPA knows one of the users with L'_0 shares his attribute private key. The suspicious user set is of course not greater than $|S|$. the TPA continues the above procedure until the suspicious set $|S|$ is not too huge. Finally, the technique for small $|S|$ can be applied and the identity in the private device can be pinpointed.

5 Analysis of the Proposed Scheme

5.1 Security Analysis

We follow the selective ciphertext-policy and chosen-plaintext (sCP-IND-CPA) attack models [24] used in most of the existing work on ABE in the literature. The goal of an adversary is to extract either the information on the message or that of the ciphertext policy.

Theorem 1 *Under the Decisional Bilinear Diffie-Hellman, Decision Linear, and q-Decisional Diffie-Hellman Inversion assumptions, the proposed scheme is sCP-IND-CPA secure and achieves user accountability.*

Proof The basic encryption algorithm in our scheme is the same with the one in [19]. We adopt the technique in [24] to extend it for multiple attribute authorities. The detail proof follows from that in [19, 24].

5.2 Performance Analysis

We only consider the time-consuming operations pairing and exponentiation. Let *Pair* be a bilinear pairing operation, *Exp* an exponentiation in $\mathbb{G}$, Exp_T an exponentiation in $\mathbb{G}_T$. We denote by n the total number of attribute values, n_k the number of attributes managed by AA_k, and ρ the bit length of the PHR user's global identity. The corresponding computation cost is $(4n + 4)Exp + Exp_T$ for PHR owners and $(\sum_{k\in[N]} 4n_k + 4\rho + 1)Pair{+}3Exp$ for PHR users.

6 Conclusion

In this paper, we propose a multi-authority CP-ABE scheme with user accountability and apply it to design an attribute-based PHR sharing system.

In the proposed solution, the access policy is hidden and hence user access privacy is protected. In particular, user accountability is achived, and thus the trust assumptions on both the authorities and the PHR users are reduced.

Acknowledgements We are grateful to the anonymous referees for their invaluable suggestions. This work is supported by the National Natural Science Foundation of China (Nos.61272455, 61100224 and 61272037), and the Natural Science Foundation of Shaanxi Province (No.2013JZ020).

References

1. Kaelber DC, Jha AK, Johnston D, Middleton B, Bates DW (2008) A research agenda for personal health records (phrs). Journal of the American Medical Informatics Association 15(6): 729–736.
2. Ibraimi L, Asim M, Petkovic M (2009) Secure management of personal health records by applying attribute-based encryption. pHealth'09, IEEE, pp 71–74.
3. Bethencourt J, Sahai A, Waters B (2007) Ciphertext-policy attribute-based encryption. SP'07, IEEE, pp 321–334.
4. Akinyele JA, Pagano MW, Green MD, Lehmann CU, Peterson ZN, Rubin AD (2011) Securing electronic medical records using attribute-based encryption on mobile devices. SPSM'11, ACM, pp 75–86.
5. Li M, Yu S, Zheng Y, Ren K, Lou W (2013) 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.
6. Chase M (2007) Multi-authority attribute based encryption. TCC'07, LNCS, vol 4392. Springer, pp 515–534.
7. Lu R, Lin X, Shen X (2013) Spoc: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency. IEEE Transactions on Parallel and Distributed Systems 24(3): 614–624.
8. Chen TS, Liu CH, Chen TL, Chen CS, Bau JG, Lin TC (2012) Secure dynamic access control scheme of phr in cloud computing. Journal of medical systems 36(6): 4005–4020.
9. Zhang R, Liu L (2010) Security models and requirements for healthcare application clouds. CLOUD'10, IEEE, pp 268–275.
10. Sun J, Fang Y (2010) Cross-domain data sharing in distributed electronic health record systems. IEEE Transactions on Parallel and Distributed Systems 21(6): 754–764.
11. Sahai A, Waters B (2005) Fuzzy identity-based encryption. EUROCRYPT'05, LNCS, vol 3494. Springer, pp 557–557.
12. Li J, Chen X, Li J, Jia C, Ma J and Lou W, Fine-grained Access Control based on Outsourced Attribute-based Encryption, In proceeding of The European Symposium on Research in Computer Security (ESORICS), LNCS 3184, pp. 592–609, 2013.
13. Li J, Kim K, Hidden attribute-based signatures without anonymity revocation. Information Sciences, 180(9): 1681-1689 , Elsevier, 2010.
14. Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. CCS'06, ACM, pp 89–98.
15. Kapadia A, Tsang PP, Smith SW (2007) Attribute-based publishing with hidden credentials and hidden policies. NDSS'07, The Internet Society, pp 179–192.
16. Nishide T, Yoneyama K, Ohta K (2008) Abe with partially hidden encryptor-specified access structure. ACNS'08, LNCS, vol 5037. Springer, pp 111–129.
17. Zhang Y, Chen X, Li J, Wong DS, Li H (2013) Anonymous attribute-based encryption supporting efficient decryption test. ASIACCS'13, ACM, pp 511–516.
18. Yu S, Ren K, Lou W, Li J (2009) Defending against key abuse attacks in kp-abe enabled broadcast systems. Securecomm'09, Springer, pp 311–329.
19. Li J, Ren K, Zhu B, Wan Z (2009) Privacy-aware attribute-based encryption with user accountability. ISC'09, Springer, LNCS, vol 5735. pp 347–362.

20. Li J, Zhao G, Chen X, Xie D, Rong C, Li W, Tang L, Tang Y (2010) Fine-grained data access control systems with user accountability in cloud computing. CloudCom'10, IEEE, pp 89–96.
21. Li J, Huang Q, Chen X, Chow SSM, Wong DS, Xie D (2011) Multi-authority ciphertext-policy attribute-based encryption with accountability. ASIACCS'11, ACM, pp 386–390.
22. Yu S, Wang C, Ren K, Lou W (2010) Attribute based data sharing with attribute revocation. ASIACCS'10, ACM, pp 261–270.
23. Boyen X, Waters B (2006) Anonymous hierarchical identity-based encryption (without random oracles). CRYPTO'06, Springer, LNCS, vol 4117. pp 290–307.
24. Chase M, Chow SS (2009) Improving privacy and security in multi-authority attribute-based encryption. CCS'09, ACM, pp 121–130.