Fine-Grained Access Control in mHealth with Hidden Policy and Traceability

Qi Li[1,2(✉)], Yinghui Zhang[3], and Tao Zhang[4]

[1] School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
liqics@njupt.edu.cn
[2] Jiangsu Key laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, China
[3] National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[4] School of Computer Science and Technology, Xidian University, Xi'an 710071, China

Abstract. Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is a well-received cryptographic primitive to securely share personal health records (PHRs) in mobile healthcare (mHealth). Nevertheless, traditional CP-ABE can not be directly deployed in mHealth. First, the attribute universe scale is bounded to the system security parameter and lack of scalability. Second, the sensitive data is encrypted, but the access policy is in the plaintext form. Last but not least, it is difficult to catch the malicious user who intentionally leaks his access privilege since that the same attributes mean the same access privilege. In this paper, we propose HTAC, a fine-grained access control scheme with partially hidden policy and white-box traceability. In HTAC, the system attribute universe is larger universe without any redundant restriction. Each attribute is described by an attribute name and an attribute value. The attribute value is embedded in the PHR ciphertext and the plaintext attribute name is clear in the access policy. Moreover, the malicious user who illegally leaks his (partial or modified) private key could be precisely traced. The security analysis and performance comparison demonstrate that HTAC is secure and practical for mHealth applications.

Keywords: CP-ABE · Partially hidden policy · Traceability · Large universe · Adaptive security

1 Introduction

Mobile Healthcare (mHealth) provides remote, on-demand, accurate health service for patients. Unlike traditional medical service, mHealth enables a patient to collect his comprehensively physical information by various wearable sensors, integrate it into the personal health record (PHR) via smart devices, and share his PHR to request health services over cloud platform. Despite its convenience,

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2019
Published by Springer Nature Switzerland AG 2019. All Rights Reserved
Q. Li et al. (Eds.): BROADNETS 2019, LNICST 303, pp. 261–274, 2019.
https://doi.org/10.1007/978-3-030-36442-7_17

mHealth service raises high security risks. The PHR contains a vast amount of private and sensitive information, including glycemic index, infectious disease, genetic history, etc. The patient expects his PHR can only be read by authorized entities. However, when the PHR data is stored on the cloud server, which may lack of effective security mechanism. Even worse, the cloud server may sell the PHR files for benefit. In such situation, the unauthorized access is unavoidable.

As a promising technique, Attribute-based encryption (ABE) [19] was designed to provide data confidentiality and fine-grained access control. Due to the fact that the ciphertext is computed from an access policy, Ciphertext-Policy ABE (CP-ABE) [2,5,22] is more suitable for the PHR owner to set the access policy for his PHR data. Although there are various ABE schemes proposed for policy expressiveness [20,23], multiple authorities [4,9,21], attribute revocation [10,24] and adaptively security [7,11], it is still worth considering the concerns of *large universe*, *policy hiding* and *traceability*.

Large Universe. In terms of the scale of attribute universe, ABE can be divided into two types: small universe and the large one. In the former ABE, the attributes should be stated in the initialization phase and the size of public parameters is usually linear with the scale of attribute universe which is polynomially bounded. If the bound is set to be too small, the system might be rebuilt as the amount of attributes exceeds such bound. If the bound is set to be too large, it will cause redundant performance. On the contrary, the scale of attribute universe in large universe schemes can be exponentially large. In addition, the size of public parameters is constant. Lewko *et al.* [8] and Rouselakis *et al.* [18] proposed a large universe ABE scheme on the composite and prime order groups, respectively. In [13], Ning *et al.* proposed an efficient large universe ABE scheme with verifiable outsourced decryption.

Policy Hiding. In traditional ABE schemes, the access policy is sent along with the ciphertext in the plaintext form. That is, any user who gets the ciphertext can obtain the access policy even if he is not authorized. Suppose that a patient defines the access policy ('Diabetes Mellitus' AND 'Doctor'), then anyone can learn the policy and infer that the patient is suffering from diabetes mellitus. The patient's privacy is entirely violated. Takashi *et al.* [15] addressed this issue and presented the first hidden policy CP-ABE scheme, where the access policy was not directly sent along with the ciphertext. Lai *et al.* [6] introduced an adaptively secure and partially hidden CP-ABE scheme, where each attribute is denoted by an attribute name and an attribute value. In [6], only the access policy with attribute names is sent along with the ciphertext, while the attribute values are embedded in the ciphertext. Compared with fully-hidden policy ABE schemes, the partially-hidden schemes can achieve a tradeoff between efficiency and fully-hidden policies. In [26], Zhang *et al.* also constructed a partially-hidden access control scheme with large universe in smart health.

Traceability. If a user abuses or leaks his private key for benefit, such malicious user must be catched to enhance the data confidentiality. However, in CP-ABE,

the user's private key is described by his attributes and multiple users may have the same attributes. It is difficult to identify who leaks his private key. To solve this issue, Liu *et al.* [12] constructed a traceable CP-ABE scheme with adaptive security and expressive access policy. Li *et al.* [17] presented a traceable and large universe CP-ABE scheme with efficient user decryption in eHealth cloud.

In this paper, we simultaneously addressed the above three main properties and proposed HTAC, a traceable access control scheme with partially-hidden access policy in mHealth. The PHR owner can define any monotonic access policies himself. In summary, our contributions are as follows:

1. *Large Universe.* There is no extra bound on the attribute universe and the size of system public parameter is constant.
2. *Policy Hiding.* Only the access policy with generic attribute names is transmitted along with the ciphertext. The sensitive attribute values are embedded in the ciphertext and unknowable to any unauthorized users.
3. *Traceability.* The source of the illegally leaked key could be precisely traced. The private PHR is protected from being abused and exposed to unauthorized access.
4. *Adaptive Security and Efficiency.* We construct HTAC on composite order groups and prove the adaptive security in the standard model. The performance analysis show that HTAC is almost as efficient as the underline scheme [26].

2 Preliminaries

2.1 Linear Secret Sharing Schemes (LSSS)

Definition 1 (LSSS [1]). *Let $\mathcal{U}$ denote the system attribute universe: $\mathcal{U} = (At_1, At_2, \ldots, At_n)$, where each attribute x is composed of two parts: the attribute name At_x and multiple attribute values. $\mathcal{VU}_x = \{\jmath_{x,1}, \jmath_{x,2}, \ldots, \jmath_{x,n_x}\}$ refers to the set of all possible values of attribute x.*

$\mathbf{A} \in \mathbb{Z}_p^{\ell \times n}$ is a share-generating matrix and ρ is a function which maps the $i-th$ row of $\mathbf{A}$ to an attribute name $At_x \in \mathcal{U}$. An LSSS is composed of two following algorithms:

- ***Secret Share:*** *It takes in a secret value $s \in \mathbb{Z}_p$ and $\mathbf{A}$. It computes the secret share $\lambda_x = A_x \cdot v$ for each row A_x of $\mathbf{A}$, where $v = (s, y_2, ..., y_n)^T$ and $y_2, ..., y_n$ are randomly chosen from $\mathbb{Z}_p$.*
- ***Secret Reconstruction:*** *It takes in the secret shares $\{\lambda_x\}$ and any authorized set $\mathcal{P}$. It sets $\mathcal{I} = \{i | \rho(i) \in \mathcal{P}\} \subseteq \{1, 2, ..., \ell\}$ and calculates the coefficients $\{\omega_i \in \mathbb{Z}_p\}_{i \in \mathcal{I}}$ such that $\sum_{i \in \mathcal{I}} \omega_i A_i = (1, 0, ..., 0)$. Then the secret s is reconstructed by $s = \sum_{i \in \mathcal{I}} \omega_i \lambda_i$.*

Similar as [7,25,26], the LSSS matrices in our scheme is constructed over $\mathbb{Z}_N$, where N is a product of multiple primes. In HTAC, the user's attribute set is denoted by $\mathcal{S} = (\mathcal{NAM}_S, \mathcal{VU}_S)$, where $\mathcal{NAM}_S \subsetneqq \mathbb{Z}_N$ is the attribute name index and $\mathcal{VU}_S = \{\jmath_{x,i}\}_{x \in \mathcal{NAM}_S}$ is the attribute value set.

In the employed access structure $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$, $\mathcal{T} = (t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(\ell)})$ refers to the set of attribute value for each row of $\mathbf{A}$. $\mathcal{S}$ satisfies $\mathbb{A}$ means that there exists $\mathcal{I} \subseteq \{1, 2, \ldots, \ell\}$ satisfying $(\mathbf{A}, \rho)$, $\{\rho(i) | i \in \mathcal{I}\} \subseteq \mathcal{NAM}_S$ and $\jmath_{\rho(i)} = t_{\rho(i)} \forall i \in \mathcal{I}$.

2.2 Composite Order Bilinear Group

A group generator $\mathcal{G}$ takes in a security parameter λ and outputs the terms $(\mathbb{G}, \mathbb{G}_1, p_1, p_2, p_3, p_4, e)$, where p_1, p_2, p_3 and p_4 are 4 different primes, the order of cyclic groups $\mathbb{G}$ and $\mathbb{G}_1$ is $N = p_1 p_2 p_3 p_4$, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ is a bilinear map with such properties:

1. Bilinearity: $\forall \varrho, \varpi \in \mathbb{G}$ and $a, b \in \mathbb{Z}_N$, we have $e(\varrho^a, \varpi^b) = e(\varrho, \varpi)^{ab}$.
2. Non-degeneracy: $\exists \varrho \in \mathbb{G}$ such that the order of $e(\varrho, \varrho)$ is N.

Denote $\mathbb{G}_{p_x}$ as the subgroup of order p_x in $\mathbb{G}$. If $\varrho_x \in \mathbb{G}_{p_x}$ and $\varrho_y \in \mathbb{G}_{p_y}$, for $x \neq y$, we have $e(\varrho_x, \varrho_y) = 1$.

2.3 Complexity Assumptions

$\mathbb{GD}$ represents the terms $(\mathbb{G}, \mathbb{G}_1, N = p_1 p_2 p_3 p_4, e)$ generated by a group generator $\mathcal{G}$.

Assumption 1. Given $\mathcal{G}$ and the following distribution:

$$g \xleftarrow{R} \mathbb{G}_{p_1}, P_3 \xleftarrow{R} \mathbb{G}_{p_3}, P_4 \xleftarrow{R} \mathbb{G}_{p_4},$$
$$\Phi = (\mathbb{GD}, g, P_3, P_4), \partial_1 \xleftarrow{R} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2}, \partial_2 \xleftarrow{R} \mathbb{G}_{p_1}.$$

The algorithm $\mathcal{A}$'s advantage in breaking this assumption is $\mathsf{Adv1}_{\mathcal{G},\mathcal{A}}(\lambda)$ $= |\Pr[\mathcal{A}(\Phi, \partial_1) = 1] - \Pr[\mathcal{A}(\Phi, \partial_2) = 1]|$.

Definition 2. *$\mathcal{G}$ satisfies Assumption 1 if $\mathsf{Adv1}_{\mathcal{G},\mathcal{A}}(\lambda)$ is negligible for any probabilistic polynomial time (PPT) algorithm $\mathcal{A}$.*

Assumption 2. Given $\mathcal{G}$ and the following distribution:

$$g, P_1 \xleftarrow{R} \mathbb{G}_{p_1}, P_2, Q_2 \xleftarrow{R} \mathbb{G}_{p_2}, P_3, Q_3 \xleftarrow{R} \mathbb{G}_{p_3}, P_4 \xleftarrow{R} \mathbb{G}_{p_4},$$
$$\Phi = (\mathbb{GD}, g, P_1 P_2, Q_2 Q_3, P_3, P_4),$$
$$\partial_1 \xleftarrow{R} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}, \partial_2 \xleftarrow{R} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}.$$

The algorithm $\mathcal{A}$'s advantage in breaking this assumption is $\mathsf{Adv2}_{\mathcal{G},\mathcal{A}}(\lambda) =$ $|\Pr[\mathcal{A}(\Phi, \partial_1) = 1] - \Pr[\mathcal{A}(\Phi, \partial_2) = 1]|$.

Definition 3. *$\mathcal{G}$ satisfies Assumption 2 if $\mathsf{Adv2}_{\mathcal{G},\mathcal{A}}(\lambda)$ is negligible for any PPT algorithm $\mathcal{A}$.*

Assumption 3. Given $\mathcal{G}$ and the following distribution:

$$g \xleftarrow{R} \mathbb{G}_{p_1}, g_2, P_2, Q_2 \xleftarrow{R} \mathbb{G}_{p_2}, P_3 \xleftarrow{R} \mathbb{G}_{p_3}, P_4 \xleftarrow{R} \mathbb{G}_{p_4},$$
$$\Phi = (\mathbb{GD}, g, g_2, g^{\alpha} P_2, g^{s} Q_2, P_3, P_4),$$
$$\partial_1 = \hat{e}(g, g)^{\alpha s}, \partial_2 \xleftarrow{R} \mathbb{G}_1.$$

The algorithm $\mathcal{A}$'s advantage in breaking this assumption is $\mathsf{Adv3}_{\mathcal{G},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\Phi, \partial_1) = 1] - \Pr[\mathcal{A}(\Phi, \partial_2) = 1]|$.

Definition 4. *$\mathcal{G}$ satisfies Assumption 3 if $\mathsf{Adv3}_{\mathcal{G},\mathcal{A}}(\lambda)$ is negligible for any PPT algorithm $\mathcal{A}$.*

Assumption 4. Given $\mathcal{G}$ and the following distribution:

$$g, h \xleftarrow{R} \mathbb{G}_{p_1}, g_2, P_2, A_2, B_2, D_2 \xleftarrow{R} \mathbb{G}_{p_2}, t', r' \xleftarrow{R} \mathbb{Z}_N$$
$$P_3 \xleftarrow{R} \mathbb{G}_{p_3}, P_4, Z, A_4, D_4 \xleftarrow{R} \mathbb{G}_{p_4},$$
$$\Phi = (\mathbb{GD}, g, g_2, g^{t'} B_2, h^{t'} Q_2, P_3, P_4, hZ, g^{r'} D_2 D_4),$$
$$\partial_1 = h^{r'} A_2 A_4, \partial_2 \xleftarrow{R} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}.$$

The algorithm $\mathcal{A}$'s advantage in breaking this assumption is $\mathsf{Adv4}_{\mathcal{G},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\Phi, \partial_1) = 1] - \Pr[\mathcal{A}(\Phi, \partial_2) = 1]|$.

Definition 5. *$\mathcal{G}$ satisfies Assumption 4 if $\mathsf{Adv4}_{\mathcal{G},\mathcal{A}}(\lambda)$ is negligible any PPT algorithm $\mathcal{A}$.*

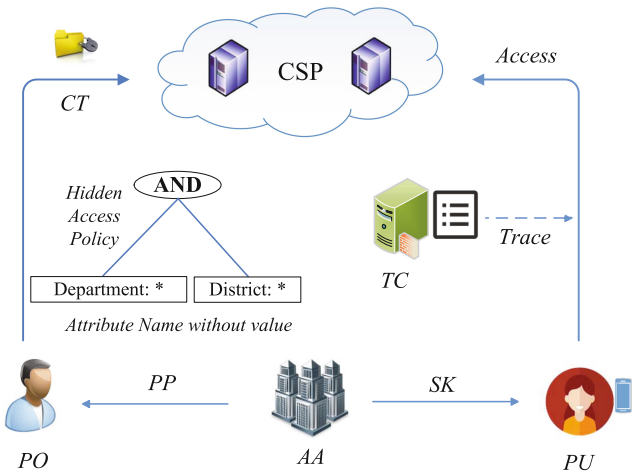

Fig. 1. System architecture of HTAC

3 System Model and Security Goals

3.1 System Model

As depicted in Fig. 1, the system architecture of HTAC consists of 5 types of entities: (1) Attribute Authority (AA), (2) Cloud Service Provider (CSP), (3) PHR Owner (PO), (4) PHR User (PU), (5) Trace Center (TC).

- AA governs the attribute universe, sets the public parameters and grants the private keys for PU according to his attributes.
- CSP stores the encrypted PHRs along with the corresponding hidden access structures. It can also delete the PHRs if necessary.
- PO integrates his own PHRs via smart devices and some wearable or implantable sensors. PO can define appropriate access structures to encrypt his PHRs before outsourcing it to CSP.
- PU is a PHR user who needs to access the encrypted PHRs to offer health care service, such as a physician. Every PU possesses some attributes and the corresponding private keys. A PU can successfully recover the encrypted PHR only if his attribute set matches the attached hidden access policy.
- TC is responsible for tracing the malicious PU who leaks his private key for some illegal purpose.

3.2 Security Goals

In HTAC, AA and TC are trustworthy. CSP is assumed to be honest-but-curious as in [24]. That is, it honestly executes the specified procedures but tries to gain secret information from encrypted PHRs. The adversary could be a malicious PU or a group of multiple PUs and CSP. Moreover, the adversary also aims to learn the attribute values of the hidden access policies from encrypted PHRs.

Concretely, we mostly focus on the security requirements as follows.

- *PHR Confidentiality.* The PHRs contains sensitive information and should be kept secrecy from any unauthorized user.
- *Collusion-Resistance.* Various malicious PUs and CSP may collude to recover the PHR ciphertext that none of them is authorized to access. HTAC should resist such collusion attacks.
- *Attribute Privacy.* In an access policy, the concrete attribute value is sensitive and should be hidden to preserve the attribute privacy.

4 Framework Definition and Security Models

4.1 Framework Definition

HTAC consists of the following five algorithms.

- $\mathsf{Setup}(\kappa, U) \rightarrow (\mathsf{PP}, \mathsf{MSK})$: By taking in a security parameter κ and the system attribute universe description U, this algorithm returns the system public parameter PP and the master key MSK. Additionally, it initializes an identity table $IT = \varnothing$.

- $\mathsf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, id, \mathcal{S}) \rightarrow \mathsf{SK}_{id,\mathcal{S}}$: This algorithm takes in PP, MSK, an identity id, and a set of attributes $\mathcal{S}$. It then returns a private key $\mathsf{SK}_{id,\mathcal{S}}$.
- $\mathsf{Encrypt}(\mathsf{PP}, \mathbb{A}, M) \rightarrow \mathsf{CT}_{\mathbb{A}}$: This algorithm takes in PP, an access structure $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$, a plaintext message M. It then returns a ciphertext $\mathsf{CT}_{\mathbb{A}}$.
- $\mathsf{Decrypt}(\mathsf{PP}, \mathsf{CT}_{\mathbb{A}}, \mathsf{SK}_{id,\mathcal{S}}) \rightarrow M$ or $\perp$: This algorithm takes in PP, $\mathsf{CT}_{\mathbb{A}}$ and $\mathsf{SK}_{id,\mathcal{S}}$. It returns M only if $\mathcal{S}$ matches $\mathbb{A}$. Otherwise, it returns $\perp$. Concretely, the decryption algorithm contains two subroutines, Matching Test and Final Decryption. If $\mathcal{S}$ does not match $\mathbb{A}$, Matching Test outputs $\perp$ to terminate the decryption process. Otherwise, it invokes the Final Decryption to return M.
- $\mathsf{Trace}(PP, \mathsf{SK}_{id,\mathcal{S}}, IT) \rightarrow id$ or $\top$. Given PP, $\mathsf{SK}_{id,\mathcal{S}}$ and IT. This algorithm first checks whether $\mathsf{SK}_{id,\mathcal{S}}$ is well-formed. If so, it returns the id associated with $\mathsf{SK}_{id,\mathcal{S}}$. Otherwise, it returns $\top$ to claim that $\mathsf{SK}_{id,\mathcal{S}}$ is not required to be traced. $\mathsf{SK}_{id,\mathcal{S}}$ is called well-formed if it can pass a 'key sanity check' [12].

4.2 CPA Security Model

The security model is described as a security game between a simulator $\mathcal{B}$ and an adversary $\mathcal{A}$.

- **Setup.** $\mathcal{B}$ runs Setup to create PP and MSK. Only PP is sent to $\mathcal{A}$.
- **Phase 1.** $\mathcal{A}$ can request the private keys of the following attribute sets $(id_1, \mathcal{S}_1)$, ..., $(id_{q_1}, \mathcal{S}_{q_1})$.
- **Challenge.** $\mathcal{A}$ gives $\mathcal{B}$ two equal-length messages M_0, M_1 and two challenge access structures $\mathbb{A}_1 = (\mathbf{A}, \rho, \mathcal{T}_0)$, $\mathbb{A}_2 = (\mathbf{A}, \rho, \mathcal{T}_1)$. $\mathcal{B}$ randomly picks $\beta \in \{0, 1\}$, computes $\mathsf{CT}_{\mathbb{A}_\beta} \leftarrow \mathsf{Encrypt}(\mathsf{PP}, M_\beta, \mathbb{A}_\beta)$ and gives $\mathsf{CT}_{\mathbb{A}_\beta}$ to $\mathcal{A}$.
- **Phase 2.** $\mathcal{A}$ requests the private keys of the following attribute sets $(id_{q_1+1}, \mathcal{S}_{q_1+1})$, ..., $(id_q, \mathcal{S}_q)$.
- **Guess:** $\mathcal{A}$ outputs its guess $\beta' \in \{0, 1\}$.

$\mathcal{A}$ wins if $\beta' = \beta$ under such restriction that neither $\mathbb{A}_1$ nor $\mathbb{A}_2$ can be matched by any queried set in Phase 1 and Phase 2. The advantage of $\mathcal{A}$ in the above game can be defined as $\left|\Pr[\,\beta' = \beta\,] - \frac{1}{2}\right|$.

Definition 6. *HTAC is adaptively secure if none polynomial time adversary can win the security game with a non-negligible advantage.*

4.3 Traceability Model

- **Setup.** $\mathcal{B}$ runs Setup to create PP and MSK. PP is sent to $\mathcal{A}$.
- **Key Query.** $\mathcal{A}$ queries the private keys of the tuples $(id_1, \mathcal{S}_1)$, $(id_2, \mathcal{S}_1), \ldots, (id_q, \mathcal{S}_q)$ from $\mathcal{B}$.
- **Key Forgery.** $\mathcal{A}$ outputs $SK_\star$. $\mathcal{A}$ wins if $\mathbf{Trace}(IT, PP, SK_\star) \neq \top$ and $\mathbf{Trace}(IT, PP, SK_\star) \notin \{id_1, id_2, ..., id_q\}$. $\mathcal{A}$'s advantage is defined as $Pr[\mathbf{Trace}(IT, PP, SK_\star) \neq \{\top\} \cup \{id_1, id_2, ..., id_q\}]$.

Definition 7. *HTAC is fully traceable if all PPT attackers have at most negligible advantage in the above game.*

5 Our Construction

5.1 Initialization

AA generates the system parameters by running the Setup algorithm.

– Setup: Firstly, AA runs the group generator $\mathcal{G}$ by taking in the system security parameter κ and obtain $\mathbb{GD} = (\mathbb{G}, \mathbb{G}_1, N = p_1p_2p_3p_4, e)$. Secondly, AA sets $\mathbb{Z}_N$ as the system attribute universe and randomly picks $\alpha, a, b \in \mathbb{Z}_N$, $g, \vartheta \in \mathbb{G}_{p_1}$, $X_3 \in \mathbb{G}_{p_3}$, $X_4, \Upsilon \in \mathbb{G}_{p_4}$ and calculates $Y = e(g,g)^{\alpha}$, $H = \vartheta\Upsilon$. Finally, $\mathsf{PP} = (g, g^a, g^b, H, Y, X_4, N)$ is published and $\mathsf{MSK} = (\vartheta, \Upsilon, \alpha)$ is kept as secret.

It also initializes the table IT to be empty.

5.2 PU Authorization

When a PU joins in the system, he will be labeled by an id and issued an attribute set $\mathcal{S} = (\mathcal{NAM}_S, \mathcal{VU}_S)$, where $\mathcal{NAM}_S \subseteq \mathbb{Z}_N$ and $\mathcal{VU}_S = \{s_i\}_{i \in \mathcal{NAM}_S}$. AA then generates the private keys for PU by running KeyGen.

– KeyGen: AA randomly picks $t \in \mathbb{Z}_N$, $c \in \mathbb{Z}_N^*$ and $R, R', R'', R_i \in \mathbb{G}_{p_3}$ for $i \in \mathcal{NAM}_S$. The private key is set as $\mathsf{SK}_{id,\mathcal{S}} = (\mathcal{S}, K, K', L, L', \{K_i\}_{i \in \mathcal{NAM}_S})$, where $K = g^{\frac{\alpha}{(b+c)}} g^{at} R, K' = g^t R', L = c, L' = g^{bt} R'', K_i = (g^{s_i}\vartheta)^{(b+c)t} R_i$.

Finally, AA records (id, c) in IT.

5.3 PHR Outsourcing

As in the KEM scheme [13], PO first employs a symmetric encryption scheme and selects a symmetric key SYK to encrypt his PHRs. Then, PO defines a hidden access policy $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$ and encrypts SYK by running Encrypt. SYK is imply set as an element in $\mathbb{G}_1$.

– Encrypt: PO randomly picks two vectors $v, \mu \in \mathbb{Z}_N^n$ where $v = (s, v_2, \ldots, v_n)$ and $\mu = (s_1, \mu_2, \ldots, \mu_n)$. Based on X_4, It also randomly chooses $Q, Q_1, Q_{\Delta,x}, Q_{c,x}, Q_{d,x} \in \mathbb{G}_{p_4}$ and $r_x \in \mathbb{Z}_N$ for $1 \le x \le \ell$. It then set $\mathsf{CT}_{\mathbb{A}} = ((\mathbf{A}, \rho), CT_T, CT_D)$, where CT_T is used for decryption test and CT_D is the real ciphertext of SYK. More specifically, CT_T is computed as $CT_T = (C_\Delta, C_{\Delta,0}, C'_{\Delta,0}, \{C_{\Delta,x}\}_{1 \le x \le \ell})$ and CT_D is calculated as $(C, C_0, C'_0, \{C_x, D_x\}_{1 \le x \le \ell})$ where $C_\Delta = Y^{s_1}, C_{\Delta,0} = g^{s_1}Q, C'_{\Delta,0} = g^{bs_1}Q_1$, $C_{\Delta,x} = g^{aA_x \cdot \mu}(g^{t_{\rho(x)}}H)^{-s_1}Q_{\Delta,x}$, $C = \mathsf{SYK} \cdot Y^s, C_0 = g^s, C'_0 = g^{bs}$ and $C_x = g^{aA_x \cdot v}(g^{t_{\rho(x)}}H)^{-r_x}Q_{c,x}, D_x = g^{r_x}Q_{d,x}$.

Finally, PO uploads the PHR ciphertext data and $\mathsf{CT}_{\mathbb{A}}$ to CSP.

5.4 PHR Access

If PU owns the appropriate attributes that match the access policy, SYK can be recovered by the running Decrypt.

– Decrypt: PU first computes $\mathbf{I}_{\mathbf{A},\rho}$, which refers to the set of minimum subsets of $\{1, 2, \ldots, \ell\}$ that satisfies $(\mathbf{A}, \rho)$. Then PU works as follows.
  - Matching Test: This algorithm checks if there exists $\mathcal{I} \in \mathbf{I}_{\mathbf{A},\rho}$ that satisfies $\{\rho(i)|i \in \mathcal{I}\} \subseteq \mathcal{NAM}_S$ and $C_{\Delta}^{-1} = \Lambda\Theta\Xi$, where $\Lambda = e\left(\prod_{i\in\mathcal{I}} C_{\Delta,i}^{\omega_i}, (K')^L L'\right)$, $\Theta = e\left(C_{\Delta,0}, \prod_{i\in\mathcal{I}} K_{\rho(i)}^{\omega_i}\right)$, $\Xi = e\left((C_{\Delta,0})^L C'_{\Delta,0}, K^{-1}\right)$ and $\sum_{i\in\mathcal{I}} \omega_i A_i = (1, 0, \ldots, 0)$ for some coefficients $\{\omega_i\}_{i\in\mathcal{I}}$. If no such $\mathcal{I}$ can be found, it returns $\perp$ to point out that $\mathcal{S}$ does not match $\mathbb{A}$. Otherwise, it runs Final Decryption by invoking the qualified $\mathcal{I}$ and $\{\omega_i\}_{i\in\mathcal{I}}$.
  - Final Decryption: This algorithm recovers SYK by computing $\mathsf{SYK} = C/B$, where

$$B = \frac{e\left((C_0)^L C'_0, K\right)}{\prod_{i\in\mathcal{I}} \left(\left(e(C_i, (K')^L L') e(D_i, K_{\rho(i)})\right)^{\omega_i}\right.}.$$

Finally, PU can use SYK to decrypt the PHR ciphertext.

5.5 Trace

To catch the malicious PU who leaks the key $\mathsf{SK}_{id,\mathcal{S}}$ in the form of $(\mathcal{S}, K, K', L, L', \{K_i\}_{i\in\mathcal{NAM}_S})$. TC can runs Trace by taking in $\mathsf{SK}_{id,\mathcal{S}}$.

– Trace: TC first calls Key Sanity Check to check if $\mathsf{SK}_{id,\mathcal{S}}$ is well-formed. If $\mathsf{SK}_{id,\mathcal{S}}$ can pass the following checks, it could be called a well-formed key. TC then searches L in IT: If L can be found, TC returns the corresponding id. Otherwise, it returns $\top$.
  - Key Sanity Check:
    (1) $L \in \mathbb{Z}_N$, $K, K', L', K_i \in \mathbb{G}$;
    (2) $e(g^b, K') = e(g, L') \neq 1$;
    (3) $e(g^b \cdot g^L, K) = e((K')^L \cdot L', g^a) \cdot e(g, g)^{\alpha} \neq 1$;
    (4) $\exists i \in \mathcal{S}, s.t. e(g^{s_i} H, (K')^L \cdot L') = e(g, K_i) \neq 1$.

6 Security Analysis

6.1 CPA Security

For simplicity, we reduce the CPA security of HTAC to that of [26]. In the following security proof, we denote the scheme [26] and HTAC be $\sum_{HAC}$ and $\sum_{HTAC}$, respectively.

Lemma 1. *$\sum_{HAC}$ is adaptively secure if Assumptions 1, 2, 3, and 4 hold.*

Proof. For the detailed security proof, please refer to [26].

Lemma 2. *$\sum_{HTAC}$ is adaptively secure in the security game of Sect. 4.2, if $\sum_{HAC}$ is adaptively secure.*

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ which can break our scheme $\sum_{HTAC}$ with advantage $ADV_{\sum_{HTAC}}$, then we can build a PPT simulator $\mathcal{B}$ to break the underline $\sum_{HAC}$ with advantage $ADV_{\sum_{HAC}}$, which is identical to $ADV_{\sum_{HAC}}$.

Setup. After receiving the public parameters $\mathsf{PP}_{\sum_{HAC}} = (g, g^a, H, Y, X_4, N)$ from $\sum_{HAC}$, $\mathcal{B}$ randomly chooses $b \in \mathbb{Z}_N$ and gives $\mathsf{PP}_{\sum_{HTAC}} = (g, g^a, g^b, H, Y, X_4, N)$ to $\mathcal{A}$. Besides, $\mathcal{B}$ initializes $IT = \varnothing$

Phase 1. When $\mathcal{A}$ queries the private key of $(id, \mathcal{S})$, $\mathcal{B}$ submits $\mathcal{S}$ to $\sum_{HAC}$ and obtains $\mathsf{SK}_{\mathcal{S}} = (\mathcal{S}, \hat{K}, \hat{K}', \{\hat{K}_i\}_{i \in \mathcal{NAM}_S})$, where $\hat{K} = g^{\alpha} g^{a\hat{t}} R, \hat{K}' = g^{\hat{t}} R', \hat{K}_i = (g^{s_i} \vartheta)^{\hat{t}} R_i$. $\mathcal{B}$ then randomly picks $c \in \mathbb{Z}_N^*$, $R'' \in \mathbb{G}_{p_3}$ and computes

$$
\begin{aligned}
&K = (\hat{K})^{\frac{1}{b+c}} = (g^{\alpha} g^{a\hat{t}} R)^{\frac{1}{b+c}} = g^{\frac{\alpha}{(b+c)}} g^{at} R^{\frac{1}{b+c}} \\
&K' = (\hat{K}')^{\frac{1}{b+c}} = (g^{\hat{t}} R')^{\frac{1}{b+c}} = g^{t} R'^{\frac{1}{b+c}} \\
&L' = (\hat{K}')^{\frac{b}{b+c}} R'' = g^{bt} R'^{\frac{1}{b+c}} R'' \\
&K_i = \hat{K}_i = (g^{s_i} \vartheta)^{\hat{t}} R_i = (g^{s_i} \vartheta)^{(b+c)t} R_i
\end{aligned}
$$

$\mathcal{B}$ then gives $\mathsf{SK}_{id,\mathcal{S}} = (\mathcal{S}, K, K', L = c, L', \{K_i\}_{i \in \mathcal{NAM}_S})$ to $\mathcal{A}$ and adds (id, c) to IT.

Remark that, $\mathcal{B}$ implicitly sets $t = \frac{\hat{t}}{b+c}$ in generating $\mathsf{SK}_{id,\mathcal{S}}$. Moreover, if $gcd(b + c, N) \neq 1$ or c has been put into IT, $\mathcal{B}$ has to randomly pick a new $c \in \mathbb{Z}_N^*$ and rebuild the private key.

Challenge. $\mathcal{A}$ gives $\mathcal{B}$ two access structures $\mathbb{A}_1 = (\mathbf{A}, \rho, \mathcal{T}_0)$, $\mathbb{A}_2 = (\mathbf{A}, \rho, \mathcal{T}_1)$ and two messages M_0, M_1 of equal length. $\mathcal{B}$ submits them to $\sum_{HAC}$ and gets the corresponding challenge ciphertext $\mathsf{CT}_{\sum_{HAC}} = (\mathbf{A}*, \rho), \hat{CT}_T = (\hat{C}_{\Delta}, \hat{C}_{\Delta,0}, \hat{C}'_{\Delta,0}, \{\hat{C}_{\Delta,x}\}_{1 \leq x \leq \ell}), \hat{CT}_D = (\hat{C}, \hat{C}_0, \hat{C}'_0, \{\hat{C}_x, \hat{D}_x\}_{1 \leq x \leq \ell}))$, where

$$
\begin{aligned}
&\hat{C}_{\Delta} = Y^{s_1}, \hat{C}_{\Delta,0} = g^{s_1} Q, \hat{C}_{\Delta,x} = g^{aA_x \cdot \mu} (g^{t_{\rho(x)}} H)^{-s_1} Q_{\Delta,x}, \hat{C} = \mathsf{SYK} \cdot Y^s, \\
&\hat{C}_0 = g^s \text{ and } \hat{C}_x = g^{aA_x \cdot v} (g^{t_{\rho(x)}} H)^{-r_x} Q_{c,x}, \hat{D}_x = g^{r_x} Q_{d,x}.
\end{aligned}
$$

$\mathcal{B}$ then sets

$$
\begin{aligned}
&C_{\Delta} = \hat{C}_{\Delta} = Y^{s_1}, C_{\Delta,0} = \hat{C}_{\Delta,0} = g^{s_1} Q, C_{\Delta,x} = \hat{C}_{\Delta,x} = g^{aA_x \cdot \mu} (g^{t_{\rho(x)}} H)^{-s_1} Q_{\Delta,x}, \\
&C = \hat{C} = \mathsf{SYK} \cdot Y^s, C_0 = \hat{C}_0 = g^s \text{ and } C_x = \hat{C}_x = g^{aA_x \cdot v} (g^{t_{\rho(x)}} H)^{-r_x} Q_{c,x}, \\
&D_x = \hat{D}_x = g^{r_x} Q_{d,x}.
\end{aligned}
$$

Additionally, $\mathcal{B}$ computes $C'_{\Delta,0} = (\hat{C}_{\Delta,0})^b$ and $C'_0 = (\hat{C}_0)^b$.

Finally, $\mathcal{B}$ gives $\mathsf{CT}_{\mathbb{A}*} = (\mathbf{A}, \rho), CT_T = (C_{\Delta}, C_{\Delta,0}, C'_{\Delta,0}, \{C_{\Delta,x}\}_{1 \leq x \leq \ell})$, $CT_D = (C, C_0, C'_0, \{C_x, D_x\}_{1 \leq x \leq \ell}))$ to $\mathcal{A}$.

Phase 2. Same as in Phase 1.

Guess. $\mathcal{A}$ returns its guess β'. $\mathcal{B}$ gives β' to $\sum_{HAC}$.

Theorem 1. *The proposed HTAC is adaptively secure if Assumptions 1, 2, 3 and 4 hold.*

Proof. The theorem follows Lemmas 1 and 2.

Table 1. Characteristic comparison with related work

Schemes	Policy hidden	Standard model	Large universe	Adaptive security	Expressiveness	Group order	Traceability
[18]	×	✓	✓	×	LSSS	Prime	×
[14]	×	✓	✓	×	LSSS	Prime	✓
[12]	×	✓	×	✓	LSSS	Composite	✓
[15]	✓	×	×	×	AND	Prime	×
[16]	✓	✓	×	×	AND	Prime	×
[6]	✓	✓	×	✓	LSSS	Composite	×
[26]	✓	✓	✓	✓	LSSS	Composite	×
Ours	✓	✓	✓	✓	LSSS	Composite	✓

Table 2. Parameter length comparison

Scheme	Public parameter	Private key	Ciphertext
[26]	$4\|\mathbb{G}\| + 1\|\mathbb{G}_1\|$	$(\|S_K\| + 2)\|\mathbb{G}\|$	$(3\|S_C\| + 2)\|\mathbb{G}\| + 2\|\mathbb{G}_1\|$
Ours	$5\|\mathbb{G}\| + 1\|\mathbb{G}_1\|$	$(\|S_K\| + 1)\|\mathbb{G}\| + 1\|\mathbb{Z}_N\|$	$(3\|S_C\| + 4)\|\mathbb{G}\| + 2\|\mathbb{G}_1\|$

6.2 Traceability

Theorem 2. *If ℓ-SDH assumption [3, 12] and Assumption 2 hold, the proposed scheme is fully traceable provided that $q < \ell$.*

Proof. We briefly introduce the proof of traceability. The simulator $\mathcal{B}$ is given two independent instances from ℓ-SDH assumption and Assumption 2. Then $\mathcal{B}$ can interact with an adversary $\mathcal{A}$ as in [12,14]. If $\mathcal{A}$ has non-negligible advantage in the traceability game, then $\mathcal{B}$'s advantage is non-negligible in breaking Assumption 2 and ℓ-SDH assumption.

6.3 Performance Comparison

Table 1 demonstrates the characteristic comparison between related works [6,12, 14–16,18,26] and ours, including access policy privacy, security model, universe scale, security level, policy expressiveness, group order and traceability. From Table 1, we can learn that our HTAC simultaneously achieves policy hidden, large universe and traceability, while the others can only realize one or two of them.

Tables 2 and 3 give the numeric performance comparison between our work and [26]. P represents a bilinear pairing. $|S_C|$, $|S_K|$ and $|I|$ represent the number of attributes associated with $\mathsf{CT}_{\mathbb{A}}$, $\mathsf{SK}_{id,\mathcal{S}}$ and $\mathcal{I}$, respectively. E and E_1 refer to an exponential operation in $\mathbb{G}$ and $\mathbb{G}_1$, respectively. From Tables 2 and 3, we learn that the size of PP, $\mathsf{SK}_{id,\mathcal{S}}$ and $\mathsf{CT}_{\mathbb{A}}$ is slightly longer than that of [26]. In the phase of encryption, matching test and final decryption, the computation cost is a little bit more than that in [26]. However, the additional parameters

Table 3. Computing cost comparison

Scheme	Encryption cost	Matching cost	Final decryption cost
[26]	$(6\|S_C\|+2)E+2E_1$	$2\|I\|E+2P$	$\|I\|E_1+(2\|I\|+1)P$
Ours	$(6\|S_C\|+4)E+2E_1$	$(2\|I\|+2)E+3P$	$2E+\|I\|E_1+(2\|I\|+1)P$

and incurred computation overhead are employed to support the traceability of HTAC, regardless of the number of involved attributes. In summary, our scheme only sacrifices tiny parameter elements and computation cost to realize traceability in comparison to the scheme [26].

7 Conclusion

We have constructed HTAC which simultaneously supported large universe, partially hidden policy and white-box traceability. The size of the system public parameters is constant. The PHR owner can define any LSSS access policy and hide the specific attribute value. None of the unauthorized users can obtain any sensitive information of attributes from the ciphertext. We proved the adaptive security in the standard model. To support the property of traceability, only a few group elements and tiny computation overhead are incurred and have no concern with the attributes.

Our future work is to alleviate the user decryption overhead by designing appreciate outsourced technique to offload the heavy matching test and decryption operations to the cloud.

Acknowledgment. This research is sponsored by The National Natural Science Foundation of China under grant No. 61602365, No. 61502248, and the Key Research and Development Program of Shaanxi [2019KW-053]. Yinghui Zhang is supported by New Star Team of Xi'an University of Posts and Telecommunications [2016-02]. We thank the anonymous reviewers for invaluable comments.

References

1. Beimel, A.: Secure schemes for secret sharing and key distribution. DSc dissertation (1996)
2. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, SP 2007, pp. 321–334, May 2007. https://doi.org/10.1109/SP.2007.11
3. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_4
4. Chase, M.: Multi-authority attribute based encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70936-7_28

5. Cheung, L., Newport, C.: Provably secure ciphertext policy ABE. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS 2007, New York, NY, USA, pp. 456–465. ACM (2007). https://doi.org/10.1145/1315245.1315302
6. Lai, J., Deng, R.H., Li, Y.: Expressive CP-ABE with partially hidden access structures. In: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2012, New York, NY, USA, pp. 18–19. ACM (2012). https://doi.org/10.1145/2414456.2414465
7. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_4
8. Lewko, A., Waters, B.: Unbounded HIBE and attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 547–567. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_30
9. Li, J., Chen, X., Chow, S.S., Huang, Q., Wong, D.S., Liu, Z.: Multi-authority fine-grained access control with accountability and its application in cloud. J. Netw. Comput. Appl. **112**, 89–96 (2018). https://doi.org/10.1016/j.jnca.2018.03.006. http://www.sciencedirect.com/science/article/pii/S1084804518300870
10. Li, Q., Ma, J., Li, R., Liu, X., Xiong, J., Chen, D.: Secure, efficient and revocable multi-authority access control system in cloud storage. Comput. Secur. **59**, 45–59 (2016). https://doi.org/10.1016/j.cose.2016.02.002. http://www.sciencedirect.com/science/article/pii/S0167404816300050
11. Li, Q., Zhu, H., Xiong, J., Mo, R., Wang, H.: Fine-grained multi-authority access control in IoT-enabled mhealth. Ann. Telecommun. **4**, 1–12 (2019)
12. Liu, Z., Cao, Z., Wong, D.S.: White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures. IEEE Trans. Inf. Forensics Secur. **8**(1), 76–88 (2013). https://doi.org/10.1109/TIFS.2012.2223683
13. Ning, J., Cao, Z., Dong, X., Liang, K., Ma, H., Wei, L.: Auditable *sigma*-time outsourced attribute-based encryption for access control in cloud computing. IEEE Trans. Inf. Forensics Secur. **13**(1), 94–105 (2018). https://doi.org/10.1109/TIFS.2017.2738601
14. Ning, J., Dong, X., Cao, Z., Wei, L., Lin, X.: White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes. IEEE Trans. Inf. Forensics Secur. **10**(6), 1274–1288 (2015). https://doi.org/10.1109/TIFS.2015.2405905
15. Nishide, T., Yoneyama, K., Ohta, K.: Attribute-based encryption with partially hidden encryptor-specified access structures. In: Bellovin, S.M., Gennaro, R., Keromytis, A., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 111–129. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68914-0_7
16. Phuong, T.V.X., Yang, G., Susilo, W.: Hidden ciphertext policy attribute-based encryption under standard assumptions. IEEE Trans. Inf. Forensics Secur. **11**(1), 35–45 (2016). https://doi.org/10.1109/TIFS.2015.2475723
17. Qi, L., Zhu, H., Ying, Z., Tao, Z.: Traceable ciphertext-policy attribute-based encryption with verifiable outsourced decryption in ehealth cloud. Wirel. Commun. Mob. Comput. **2018**, 1–12 (2018)
18. Rouselakis, Y., Waters, B.: Practical constructions and new proof methods for large universe attribute-based encryption. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS 2013, New York, NY, USA, pp. 463–474. ACM (2013). https://doi.org/10.1145/2508859.2516672

19. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_27
20. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_4
21. Xue, K., Xue, Y., Hong, J., Li, W., Yue, H., Wei, D.S.L., Hong, P.: RAAC: robust and auditable access control with multiple attribute authorities for public cloud storage. IEEE Trans. Inf. Forensics Secur. **12**(4), 953–967 (2017). https://doi.org/10.1109/TIFS.2016.2647222
22. Yang, K., Jia, X., Ren, K.: Secure and verifiable policy update outsourcing for big data access control in the cloud. IEEE Trans. Parallel Distrib. Syst. **26**(12), 3461–3470 (2015). https://doi.org/10.1109/TPDS.2014.2380373
23. Yang, Y., Liu, X., Deng, R.H.: Lightweight break-glass access control system for healthcare internet-of-things. IEEE Trans. Ind. Inform. **14**, 3610–3617 (2017). https://doi.org/10.1109/TII.2017.2751640
24. Yu, S., Wang, C., Ren, K., Lou, W.: Achieving secure, scalable, and fine-grained data access control in cloud computing. In: IEEE INFOCOM 2010 Proceedings, pp. 1–9, March 2010. https://doi.org/10.1109/INFCOM.2010.5462174
25. Zhang, L., Hu, G., Mu, Y., Rezaeibagha, F.: Hidden ciphertext policy attribute-based encryption with fast decryption for personal health record system. IEEE Access **7**, 33202–33213 (2019). https://doi.org/10.1109/ACCESS.2019.2902040
26. Zhang, Y., Zheng, D., Deng, R.H.: Security and privacy in smart health: efficient policy-hiding attribute-based access control. IEEE Internet Things J. **5**(3), 2130–2145 (2018)