Multi-authority Access Control with Anonymous Authentication for Personal Health Record

Leyou Zhang, Yadi Ye, and Yi Mu (Senior Member, IEEE)

***Abstract*—A personal health record (PHR) system is a smart health system that serves patients and doctors. A PHR is usually stored in a cloud and managed by a semi-trusted cloud provider. However, there is still a possibility of the exposure of personal health information to semi-trusted parties and unauthorized users. To protect the privacy of patients and ensure that patients can control their PHRs, a patient-centric PHR sharing framework is proposed in this paper. In this framework, all PHRs are protected with multi-authority attribute-based encryption before outsourcing, which solves the key hosting problem and achieves fine-grained access control to PHRs. Furthermore, an anonymous authentication between the cloud and the user is proposed to ensure data integrity on the cloud while not exposing the user's identity during authentication. The proposed authentication is issued from a new online-offline attribute-based signature. It can make the encrypted PHRs resist collusion attacks and not be forged during the period of sharing, which enhances patients' control to their PHRs. Online-offline and outsourcing decryption also reduces calculation costs and improves operational efficiency. Finally, comparisons are given based on numerical experiments.**

***Index Terms*—Anonymous authentication, attribute-based signature, multi-authority attribute-based encryption, personal health record**

I. INTRODUCTION

IN recent years, as an emerging technology, PHR has played a crucial role in data sharing. PHR can store medical records online and be accessed by patients and their doctors anytime, anywhere. However, when data sharing is implemented, PHR also brings problems such as privacy leakage. In order to protect the privacy of patients and enhance the control to their PHR, the fine-grained access control scheme over sharing data based on attribute-based encryption(ABE) is proposed [1], [2], [5] and has been a hot topic at present.

ABE defines an access policy through attributes associated with generating the private key or ciphertext and only users whose attribute sets satisfy the access policy can access PHR. However, some previous schemes used a single center to generate keys and authenticate users, which undoubtedly overburdened the system. Multi-authority encryption scheme requiring multiple authorities to jointly generate private keys for users solves such problem [6]. Wei et al. [18] realized secure and efficient access control in a multi-authority environment, but the user's fuzzy authentication poses a threat to data security. In order to further ensure security, adding a searchable public key encryption scheme to a PHR system was presented [3] and authentication technology was introduced to connect users of medical system to other trusted users [7]. At the same time, some feasible solutions also effectively solve the problem of patient's privacy leakage and the confidentiality of the scheme. In these methods, the user's sensitive information, such as identity and attributes, is hidden during the system interaction [9]. For access policy containing the sensitive information of the users, hiding the access control policy is also considered in recent works [10]–[12]. However, all of them are based on sacrificing efficiency. Online and offline technology enables users to quickly obtain the final ciphertext, which decreases the computation cost and brings great convenience for users [13]. But it also brings many problems to use multi-authority attribute-based encryption in a PHR, such as anonymous authentication outsourcing and ciphertext unforgeability, users and authorities collusion etc.

This work was partly supported by the National Nature Science Foundation of China under Grant 61872087, the National Cryptography Development Fund under grant No.MMJJ20180209, International S&T Cooperation Program of Shaanxi Province No.2019KW-056, and Key Foundation of National Natural Science Foundation of China under grant NO.U19B2021.

Leyou Zhang and Yadi Ye are school of Mathematics and Statistics, Xidian University, Xi'an 710126,China.

Yi Mu is with the Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350007, China.

II. MOTIVATION

A PHR can provide convenient data storage and sharing. However, sensitive users' data on cloud devices may be stolen by unauthorized users, which causes serious privacy leakage issues. Patients will lose direct control over their PHRs. Therefore, a PHR system stored in the cloud suffers more external and internal attacks than a paper-based PHR. It is essential to provide a secure, privacy-protected PHR system with fine-grained access control. For example, a data owner defines an access policy, as shown in Fig. 1, then encrypts a PHR and saves its ciphertext to the cloud.

A promising approach is to encrypt patients' confidential data to ensure security and privacy before outsourcing it to the cloud. It is worth noting that patients should be able to decide with which users to share their PHRs. As shown in Fig.2, only users who have the corresponding key can access the encrypted PHR. However, current schemes cannot guarantee data confidentiality, ciphertext unforgeability or users' privacy in a PHR. First, patients' identities and attributes should not be disclosed during access and authentication. Second, the ciphertext encrypted by the data owner should not be tampered with by the untrusted cloud. Third, during encryption and decryption, the amount of calculation on the client side should be minimized.

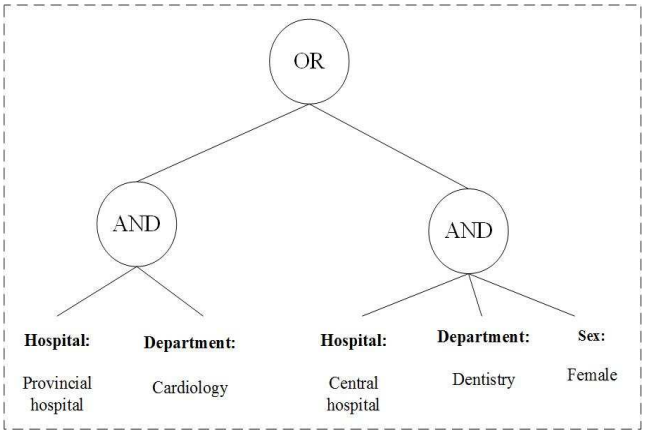

Fig. 1. Example of Access Policy

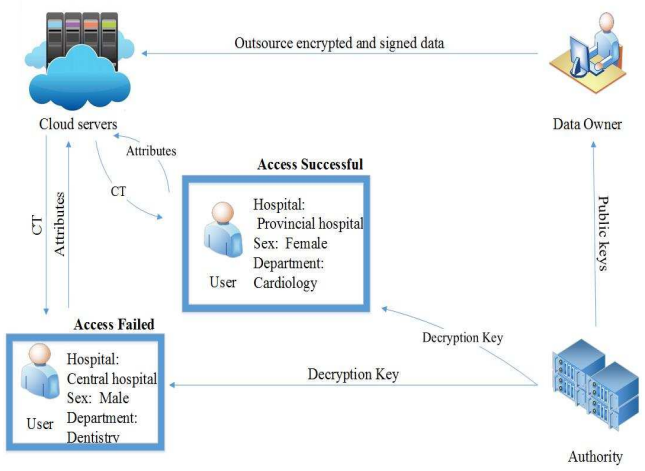

Fig. 2. Example of Access Control

A. Our Contributions

To solve the above problems, we propose a scheme that improves security and privacy in multi-authority attribute-based encryption with anonymous authentication outsourcing. The contributions of this article mainly include the following three aspects.

1. In previous schemes, users submit their identities or attribute sets to the cloud server to get the ciphertext, which gives the cloud confidential information from users. The cloud may replace or tamper with the initial ciphertext by responding with a forged transformation, and it can also deceive users by responding a terminator $\perp$. Aiming at these problems, we propose a framework by combining attribute-based encryption with attribute-based signature to better the trade-off between protecting users' privacy and guaranteeing the data security.

2. An anonymous authentication between the cloud and the user is proposed, which guarantees the data integrity in cloud and data can not be forged. In addition, anonymity of the protocol keeps user's identities not to be exposed during authentication, which achieves the privacy-preserving of users.

3. To achieve the lightweight computation, we use offline-online technique and outsourcing decryption operations to help with authentication and partial decryption.

B. Paper Organization

The following is the organization of the rest of this article. In section III, we summarize and discuss related work. Preliminaries are introduced in section IV. Section V provides an overview, including the system model, algorithm definition, security model and design goal. Section VI explains the construction of our scheme. Section VII introduces security analysis. Performance analysis is discussed in Section VIII and Section IX relates our conclusions.

III. Related Work

A. Attribute-based Encryption

Attribute-based encryption (ABE) was first proposed by Sahai and Waters [14] to solve the problem of access authorization to data outsourced to the cloud. They demonstrated the flexibility of encryption policies and the granularity of access control, accelerating security applications in outsourced data systems. In ABE, a patient defines an access policy to encrypt his/her PHR, and when a user's attributes meet the access policy, PHR can be successfully accessed. In 2006, Goyal et al. [15], [16] divided ABE into KP-ABE and CP-ABE depending on whether the access policy exists in the secret key or ciphertext. But these schemes rely on a single center to generate all the parameters of the private key from one entity, which causes a secret key escrowing problem. Then multi-authority attribute-based encryption was introduced in 2007 [17] to solve the above question, where many authorities manage different attributes, and the workload is distributed over them. In 2018, Wei et al. [18] proposed secure and effective access control in a multi-privilege environment. However, a user's access control implements fuzzy authentication between the cloud and users, which poses a threat to data security. In 2019, Yan et al. [35] proposed a multi-authority attribute-based encryption scheme with policy updating to protect attribute privacy, where the patient's attributes are composed of attribute name and attribute value. However, the decryption step requires a lot of calculation, resulting in a large consumption of resources. Li et al. [37] achieved public verifiability of ciphertext in a multi-center environment, which made the system more secure and more efficient. However, the user's private information should be further protected.

B. Anonymous Authentication

In this protocol, both messages and identities are authenticated. Message authentication determines whether a message has been forged or tampered with. Identity authentication validates the sender of a message.

Attribute-based signature (ABS) scheme can be used to issue an anonymous authentication mechanism in cloud sharing scheme. An ABS [19], [27], [29], [30] in an attribute-based system extends IBS [28] to represent a signer's identity by a set of attributes and provides end-to-end secure communication. ABS assures the verifier that the attribute set of a signer satisfies complex predicate represents the supporting message. However, the signature time of ABS increases with the complexity of the validated predicate, which causes a

huge computational burden. Zhang et al. [20] introduced an offline-online signature scheme, which improves efficiency by taking much of the computation offline. But it generates private keys from a single center, which causes key escrow issues. Ruj et al. [21] introduced a scheme by which the cloud performs a series of identity verifications before storing data, without knowing the identity. Li et al [22] introduced a decentralized authentication and revocation scheme with a blinding factor. Liu et al. [23] and Nicanfar et al. [24] used authentication technology to enhance security. In 2019, Belguith et al [38] proposed an accountable ABS to protect user's privacy. If necessary, the organization can display the identity of anonymous authenticated users. But at the same time, the low computational efficiency is a defect of their schemes. Additionally, most of the others also bear a large computing burden. Our scheme utilizes the online-offline technology to achieve a lightweight authentication, while hiding the user's identity and attributes for privacy-preserving. Furthermore, outsourced partial decryption also improves the computational efficiency of the proposed scheme.

IV. Preliminaries

A. Bilinear Groups

Define G_0 and G_T as two cyclic groups whose prime order is p. g is a generator of G_0. Let e be a bilinear map such that e: $G_0 \times G_0 \rightarrow G_T$. The map's properties has three aspects:

1. Bilinearity: $e(t_1^a, t_2^b) = e(t_1, t_2)^{ab}$, where $t_1, t_2 \in G_0$ and $a, b \in Z_p$;

2. Non-degeneracy: $e(g, g) \neq 1$;

3. Computability: G_0 and e can be computed efficiently. It is worth noting that the map e is symmetric because $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

B. Linear Secret-sharing Scheme (LSSS)

A linear secret scheme $\prod$ is linear over Z_p if

1. The shared value of each attribute about the secret value $s \in Z_p$ constitutes a vector.

2. For every access structure of the attribute set, there are a matrix and functions. The function maps the row number of the matrix to the attribute. s about the shares of the $\prod$ is assigned to attributes. Moreover, $\prod$ satisfies the linear reconstruction property. The detailed definition refers to [26].

C. Decisional q-parallel Bilinear Diffie-Hellman Exponent (q-PBDHE)

Define $s, a, b_1, b_2, \cdots, b_q$ as random elements of Z_p. Let $e : G_0 \times G_0 \rightarrow G_T$ be a bilinear map. g is a generator of G. Given a tuple

$$\vec{Y} = g, g^s, g^a, \cdots, g^{a^q}, g^{a^{q+2}}, \cdots, g^{a^{2q}}, g^{s \cdot b_j}, g^{\frac{a}{b_j}}, \cdots, g^{\frac{a^q}{b_j}}, g^{\frac{a^{q+2}}{b_j}}, \cdots, g^{\frac{a^{2q} \cdot b_k}{b_j}}, g^{\frac{a \cdot s \cdot b_k}{b_j}}, g^{\frac{a^2 \cdot s \cdot b_k}{b_j}} \cdots, g^{\frac{a^q \cdot s \cdot b_k}{b_j}}$$

$$(\forall 1 \leq j, k \leq q, k \neq j).$$

if an adversary $\mathcal{A}$ can not distinguish $e(g, g)^{a^{q+1} \cdot s}$ from a random element R of G_T in the probabilistic polynomial time, then the decisional $q - PBDHE$ assumption holds.

D. Computational Diffie Hellman Problem (CDH)

Given a generator g of order n, given two group elements $g^a \in G$ and $g^b \in G$, where a and b are two secrets. The problem of generating g^{ab} from g^a and g^b is called CDH problem.

E. Pseudo-random Function

A pseudo-random function (PRF) is a pseudo-random number generator that can obtain secure pseudo-random numbers through an operation. These distributions are indistinguishable from uniform distributions against all polynomial bounded black box attacks. That is, we can obtain its value by entering a random number into the function. For example, a user desiring to calculate $f(x)$ can send the number x to the server's member and receive information enabling the operation.

V. Overview

Following is an overview of the proposed scheme.

A. System Model

Figure. 3 shows our scheme's interactions. There are six roles in this solution, and their roles are described in detail below.

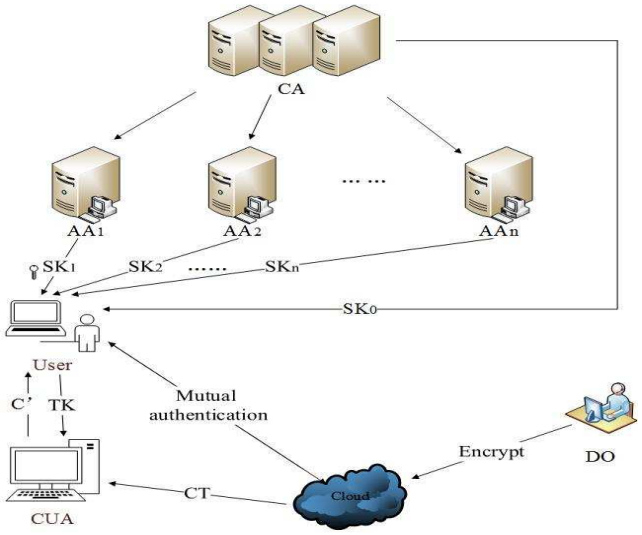

Fig. 3. System Model

Central authority (CA): CA is a trust entity that generates the system public parameters and master secret key. It also generates partial secret keys and anonymous identity credentials for users, along with the public signing key and corresponding private key. Then CA sends anonymous identity credentials and partial secret keys to users, and the signing private key to the data owner.

Attribute authority (AA): Each AA is responsible for managing attributes that are disjoint from other AAs, and publishing attributes to data owner to generate access policies to encrypted data. AAs produce local public parameters and secret keys. They are responsible for users to generate attribute secret keys.

User (U): U obtains the AIC and partial secret key from the CA and sends it to the AA. After authentication, the AA generates the attribute private key for U. If the U's identity and the ciphertext are not tampered with, then U sends the signing secret key and transformation key to cloud user assistant and decrypts the ciphertext by getting retrieving keys.

Data owner (DO): The DO owns the data and shares them by outsourcing them to the cloud server. The DO first defines the expected access policy on the attributes and then encrypts the data by calling the proposed scheme. Then the ciphertext is signed by using ABS and uploaded to the cloud.

Cloud Server (CS): The CS stores ciphertext signed by the DO. We assume that it is not trusted, i.e, CS may replace or tamper with ciphertext generated by the DO with fake conversions, and it may deceive U by returning a terminator.

Cloud user assistant (CUA): CUA running on the cloud checks the unforgeability of ciphertext and realizes its partial decryption for the user. Most computation is transferred to the CUA, which reduces the computation cost on the user side.

B. Algorithm definition

Setup $(1^{\lambda}) \rightarrow (PK, PP, MSK, SSK)$*:* The setup algorithm takes in the security parameter λ and generates public key PK, public parameters PP, master secret key MSK, and signing secret key SSK.

Authorities setup (PP)$\rightarrow (PK_j, SK_j)$*:* Authorities use PP to receive their public keys PK_j and secret keys SK_j.

User registration (u, GID)*→(AIC):* The user sends attribute set u and unique identity GID to CA, and CA outputs AIC.

Key generation $(AIC, MSK, PK,\ PK_j) \rightarrow (sk_{ue}, sk_{us})$*:* The algorithm takes AIC, MSK, PK, and PK_j and outputs transformation private key TK_{ue}, retrieving key RK_{ue} and signing private key sk_{us}.

Encrypt $(PK, m,\ (\hat{M}, \rho))$*→(CT):* The algorithm uses PK and access policy $(\hat{M}, \rho)$ to encrypt the message m, and outputs ciphertext CT.

Offline.sign $(SSK, M, PK) \rightarrow (\sigma^{'})$*:* The offline algorithm inputs PK, SSK, and the signing policy M, and outputs signature $\sigma^{'}$.

Online.sign $(CT, \sigma^{'}) \rightarrow$*(C):* The algorithm takes as input SSK, CT, and $\sigma^{'}$, and produces σ.

Decrypt (SPK, SK, CT)$\rightarrow$ *m or* $\perp$*:* The algorithm uses SPK, SK, and CT to recover m. If U is valid, then the algorithm returns the message m. Otherwise, the user gets nothing.

TABLE I
EXPLANATION OF SYMBOLS

Notation	Description
C_0, g^{ν}	system public key for encryption and signature
PK_j	AA_j's public key for encryption
Γ_j, Ω_j	AA_j's public keys for signature
SK_j	AA_j's private key for encryption
SSK	signing master key
TK_{ue}, RK_{ue}	transformation secret key and retrieving secret key for U
SK_{us}	signing secret key for U
PN_{uj}	pseudonym generated by AA_j
$sig_{a_{ij}}$	signature for attribute a_{ij}
AIC	anonymous identity credential
$\widehat{AA_j}$	attribute set managed by AA_j
Λ_{AA_j}	the name of attribute authority AA_j
$\sigma^{'}, \sigma$	offline signature, online signature
C	DO's signature on ciphertext

C. Security Model

Confidentiality

The proposed scheme proves to be confidential by using a selective access structure model.

Setup: The challenger $\mathcal{D}$ randomly picks c_0, v by running the *Setup* and *AuthoritySetup* algorithms, and then PP, PK are sent to the attacker $\mathcal{A}$. $\mathcal{A}$ sends a challenge access policy $\mathbb{A}^*$ to $\mathcal{D}$.

Phase 1: For user U with unique identity GID, $\mathcal{A}$ adaptively makes queries about secret keys corresponding to u. It is noted that u does not satisfy the access policy $\mathbb{A}^*$. $\mathcal{D}$ runs the *User registration* algorithm to generate AIC for U. After successfully authenticating the identity of U, $\mathcal{D}$ sends secret keys to $\mathcal{A}$ according to *Key generation*.

Challenge: $\mathcal{A}$ chooses two equal-length messages m_0, m_1, and $\mathcal{D}$ generates a challenge ciphertext according to the challenge access policy $\mathbb{A}^*$. $\mathcal{D}$ picks a bit $\theta \in \{0, 1\}$ and outputs the challenge ciphertext CT^*.

Phase 2: It operates the same as *Phase 1*.

Guess: $\mathcal{A}$ outputs a guess $\theta^{'}$. If $\theta^{'}$ is equal to θ, then $\mathcal{A}$ wins this game.

Definition 1: If the adversary does not win the above game with a non-negligible advantage in polynomial time, then our solution is *q-PBDHE* security.

Unforgeability

Our solution meets the requirement of unforgeability, which has a query phase and a forgery query phase.

Query phase: An adversary $\mathscr{A}$ performs three queries to challenger $\mathcal{D}$: signing key query, offline signature query, and online signature query.

Signing key query:

(1) $\mathscr{A}$, whose attribute set is u_0^*, selects an identity GID^*;

(2) $\mathscr{A}$ asks the challenger $\mathcal{D}$ for its attributes signature key. If the list F maintained by CA contains the key, then the corresponding signing key sk_{us}^* is sent by $\mathcal{D}$ to $\mathscr{A}$. Otherwise, $\mathcal{D}$ runs the *key generation* algorithm and returns the resulting signing key sk_{us}^* to $\mathscr{A}$. Then $\mathcal{D}$ adds sk_{us}^* to the list F.

Offline signature query:

(1) The adversary $\mathscr{A}$ randomly chooses a message $\mathcal{X}$ and challenge access structure $\mathbb{A}$;

(2) $\mathscr{A}$ performs some queries on the signature of challenger $\mathcal{D}$;

(3) $\mathcal{D}$ returns $\sigma^{'}$ by executing the algorithm.

Online signature query: $\mathcal{D}$ generates a valid signature σ based on $\sigma^{'}$.

Forgery phase: The adversary $\mathscr{A}$ returns a message $\mathcal{X}^*$ and a signature σ^*. $\mathscr{A}$ can win the game if

(1) $\mathscr{A}$ performs no signature queries on $(\mathbb{A}, \sigma)$ during the online signature;

(2) For the challenge access structure $\mathbb{A}$, $u^* \subseteq \mathbb{A}$;

(3) Check $\mathcal{X}^*$ is valid,

The winning advantage of $\mathscr{A}$ is

$$Adv_\lambda(\mathscr{A})(\lambda) = Pr[\mathscr{A}].$$

If the probability that the adversary winning the game in polynomial is not negligible, then this scheme is unforgeable.

Definition 2: If the adversary's advantage of $Adv_\lambda(\mathscr{A})(\lambda)$ is negligible in polynomial time, then the proposed scheme is unforgeable against chosen message attack.

D. Design Goal

Message confidentiality: If the attribute set of a user does not meet the access policy generated by the DO, then the user cannot access the message.

Ciphertext unforgeability: Only valid users with attribute sets that satisfy the access policy can access the message. If the ciphertext is tampered with by the cloud, then it cannot be verified successfully by CUA.

User's identity and attribute privacy: To protect the privacy of the user in this process, only the trusted CA knows the identity and attributes of the user.

Collusion resistance: Colluders cannot collude with others to obtain a plaintext message because they cannot complete accurate authentication between CSP and U.

VI. Construction

The proposed scheme includes eight algorithms.

Setup(1^λ): CA executes Algorithm 1. With input of security parameter λ, Algorithm 1 returns public parameter PP.

Algorithm 1: Setup

Input: a security parameter λ
Output: public parameter PP
1. Choose group G_0 and G_1 with prime order p, and a bilinear mapping: $e : G_0 \times G_0 \to G_1$;
2. Select hash functions: $H : \{0,1\}^* \to G_0$, $H_1 : \{0,1\}^* \times G_0 \to \{0,1\}^*$, $H_2 : G_0 \times \{0,1\}^* \times \{0,1\}^* \to G_0$;
3. Choose group elements $g, g_2, f_1, \cdots, f_n \in G_0$ randomly;
4. Choose exponent $c_0 \in Z_p$ at random, and generate public-secret key pair (C_0, c_0), where $C_0 = e(g,g)^{c_0}$ is the relevant public key and c_0 is the master secret key;
5. Pick $\alpha, \gamma \in Z_p$ to generate signing public keys: $\Gamma_j = f_j^\alpha$, $\Omega_j = f_j^\gamma$, and the signing master key is $SSK = (\alpha, \gamma)$;
Return
$PP = \{g, g_2, \{f_j, \Gamma_j, \Omega_j\}_{j\in[1,n]}, p, G_0, G_1, g^\nu, C_0, H, H_1, H_2, \}$

Algorithm 2: Authority Setup

Input: PP
Output: authority's public key PK_j and secret key SK_j
for j from 1 to n **do**
1. Choose β_j, $c_j \in Z_p$, and compute $C_j = g^{c_j}$;
2. Pick y_{ij} at random for each attribute $a_{ij} \in \widehat{AA_j}$, calculate $Y_{ij} = g^{y_{ij}}$;
end
Return
$PK_j = \{C_j, (Y_{ij})_{a_{ij}\in\widehat{AA_j}}\}, SK_j = \{\beta_j, c_j, (y_{ij})_{a_{ij}\in\widehat{AA_j}}\}$

Authorities setup: Every attribute authority AA_j runs Algorithm 2 to get public key PK_j and secret key SK_j.

User register: As shown in TABLE II, U, with attribute set u, provides the unique identity GID to CA when making registration. CA runs Algorithm 3 to blind GID: $T = H(GID)$ and generates the corresponding pseudonym $PN_{uj} = H_1(T \parallel r_{uj})$. For every $a_{ij} \in u$, the signature $sig_{a_{ij}}$ and AIC are generated by Algorithm 3. CA sends all signatures $\{sig_{a_{ij}}\}_{a_{ij}\in u, j\in[1,n]}$ and anonymous identity credential AIC to U.

Algorithm 3: User Register

Input: attribute set u and GID
Output: AIC
1. Draw $\kappa_{base} \in G_0$ at random;
2. Blind GID: $T = H(GID)$;
for j from 1 to n **do**
select $r_{uj} \in G_0$ and generate the corresponding pseudonym $PN_{uj} = H_1(T \parallel r_{uj})$;
for $a_{ij} \in u$, pick $\nu \in Z_p$, publish g^ν and compute the signature $sig_{a_{ij}} = H_2(Y_{ij}, PN_{uj}, \Lambda_{AA_j})^v$;
end
Return $AIC = \{Y_{ij}, PN_{uj}, \Lambda_{AA_j}, sig_{a_{ij}}\}_{a_{ij}\in u, j\in[1,n]}$

TABLE II
User registration

User		CA
1. u,GID	$\rightarrow$	
		2. $T = H(GID)$, $PN_{uj} = H_1(T \parallel r_{uj})$ $sig_{a_{ij}} = H_2(Y_{ij}, PN_{uj}, \Lambda_{AA_j})$, $AIC = \{PN_{uj}, \Lambda_{AA_j}, sig_{a_{ij}}\}_{a_{ij}\in u, j\in[1,n]}$
	$\leftarrow$	
3. $sig_{a_{ij}}$,AIC		

Key generation: The interactive process of the algorithm is depicted in TABLE III and TABLE IV. U sends $\{Y_{ij}, PN_{uj}, \Lambda_{AA_j}, sig_{a_{ij}}\}_{a_{ij}\in u\cap\widehat{AA_j}}$, U's parts of AIC to AA_j. Then, for every $a_{ij} \in u \cap \widehat{AA_j}$, AA_j executes the Algorithm 4 to return the corresponding partial private key if U's identity is passed. CA stores all the seeds of $PRF(\cdot)$, such as β_j generated by $AA_j, j \in [1,n]$. These seeds are transmitted in a secure channel.

CA obtains β_0 and k_0 through Algorithm 5. Finally, U gets the transformation key

$$TK_{ue} = \{e, k_0, (k_j, p_j)_{j\in[1,n]}\}.$$

and retrieving key $RK_{ue} = e$. CA also executes Algorithm 5 to output the signing private key SK_{us} and sends it to DO in a secure channel.

Algorithm 4: Key Generation 1

Input: $AIC, Y_{ij}, PN_{uj}, \Lambda_{AA_j}, sig_{a_{ij}}$
Output: $k_j, p_j, (F_{ij})_{a_{ij}\in u\cap\widehat{AA_j}}$
if
$e(sig_{a_{ij}}, g) = e(H_2(Y_{ij}, PN_{uj}, \Lambda_{AA_j}), g^{\nu})$;
for j from 1 to n **do**
Choose $w_{uj} \in Z_p$, compute $\beta_j^{'} = PRF_{\beta_j}(PN_{uj}), p_j = g^{w_{uj}}$, $k_j = g^{\beta_j^{'}} g^{c_j w_{uj}}$, $(F_{ij} = Y_{ij}^{w_{uj}})_{a_{ij}\in u\cap\widehat{AA_j}}$;
end
return $(k_j, p_j, (F_{ij})_{a_{ij}\in u})_{j\in[1,n]}$
else return $\perp$.

Algorithm 5: Key Generation 2

Input: $\beta_j^{'}$
Output: RK_{ue} and SK_{us}
Select $e \in Z_p$ randomly;
Compute $\beta_0 = c_0 - \sum \beta_j^{'}$, calculate $k_0 = g^{\beta_0} g^e, T_i = \kappa_{base}^{\frac{1}{\alpha+\gamma i}}$;
Return $RK_{ue} = e, SK_{us} = \{\kappa_{base}, \{T_i \mid i \in u\}\}$

TABLE III
KEY GENERATION

User	AA_j
1.$AIC_{a_{ij}\in u\cap\widehat{AA_j}}$ →	
	2. Check $e(sig_{a_{ij}}, g) = e(H_2(Y_{ij}, PN_{uj}, \Lambda_{AA_j}), g^{v})$. If true, $\beta_j^{'} = PRF_{\beta_j}(PN_{uj})$, $k_j = g^{\beta_j^{'}} g^{c_j w_{uj}}, p_j = g^{w_{uj}}$, $(F_{ij} = Y_{ij}^{w_{uj}})_{a_{ij}\in u\cap\widehat{AA_j}}$
←	
3.$k_j, p_j, (F_{ij})_{a_{ij}\in u\cap\widehat{AA_j}}$	

Encrypt: DO chooses a message $m \in G$, access structure $(\hat{M}, \rho)$, and vector $\vec{v} = (s, v_2, \cdots, v_n)$, where $s, v_2, \cdots, v_n \in Z_p$ are randomly selected, and $\hat{M}$ is an $\sum_{j=1}^{n} l_j \times n$ matrix. It computes $\lambda_{ji} = \hat{M}_j^i \vec{v}$, which is only made up of rows about attributes monitored by AA_j from $\hat{M}$. ρ_i means the attribute labeled row i. DO runs Algorithm 6 to get the ciphertext CT. Then DO signs CT.

Offline.sign: Let π represent the mapping from rows to the attributes. M_x denotes row x of M, where $M \in Z_p^{l\times n}$. So, $\pi(x)$ maps from M_x to attribute x. DO executes Algorithm 7 to generate $\sigma^{'}$.

Online.sign: Input $\{\sigma^{'}, CT, T\}$, $\mu = H(CT \parallel T)$. Algorithm 8 computes the signature σ. DO sends $C = (CT, \tau, \sigma)$ to the cloud.

TABLE IV
KEY GENERATION

User	CA
	1. Computes $\beta_j^{'} = PRF_{\beta_j}(PN_{uj})$, $\beta_0 = c_0 - \sum_{i=1}^{n} \beta_j^{'}, k_0 = g^{\beta_0} g^e$ $T_i = \kappa_{base}^{\frac{1}{\alpha+\gamma i}}, i \in u$
←	
2. Obtain k_0, e, and $SK_{us} = \{\kappa_{base}, T_i \mid i \in u\}$	

Algorithm 6: Encrypt

Input: PK, m and $(\hat{M}, \rho)$
Output: CT
Select vector $\vec{v} = (s, v_2, \cdots, v_n)$, compute $C_1 = m \times e(g, g)^{c_0 s}$, $S = g^s$;
for j from 1 to n **do**
Select $r_{j1}, r_{j2}, \cdots, r_{jl_j} \in Z_p$, compute
$(C_{j1} = g^{c_j \lambda_{j1}} Y_{\rho_j(1)}^{-r_{j1}}, D_{j1} = g^{r_{j1}}), \cdots,$
$(C_{jl_j} = g^{c_j \lambda_{jl_j}} Y_{\rho_j(l_j)}^{-r_{jl_j}}, D_{jl_j} = g^{r_{jl_j}})$;
end
Return $CT = \{C_1, S, (C_{jr}, D_{jr})_{r\in[1,l_j], j\in[1,n]}\}$

Algorithm 7: Offline.sign

Input: SK_{us}, N, PP, PK
Output: $\sigma^{'}$
Compute v_i which satisfies $\sum_{i=1}^{l} N v_i = (1, 0, \ldots, 0)^T$, choose t_0, $t_i \in Z_p$, $i \in u$, $N \leftarrow (Z_p^{l\times n})$;
Compute $Y = \kappa_{base}^{t_0}, Q_i = (T_i^{v_i})^{t_0} (g_2 g^{\lambda})^{t_i}$;
for j from 1 to n **do**
$P_j = \prod_{i=1}^{l} (f_j^{\alpha} f_j^{\gamma i})^{N_{ij} t_i}$;
end
Return $\sigma^{'} = \{Y, \{Q_i\}_{i\in[1,l]}, \{P_j\}_{j\in[1,n]}, N\}$

Algorithm 8: Online.sign

Input: $\{\sigma^{'}, CT, T\}$, $\mu = H(CT \parallel T)$
Output: σ, C
Choose τ as time stamp;
Compute $S_i^{'} = (v_i - v_{\pi(i)}) t_0, S_i^{''} = t_i - t_{\pi(i)}, S_i^{'''} = \mu t_i - \lambda t_{\pi(i)}$, $P_{ij}^{'} = (M_{ij} - N_{ij}) t_i, P_{ij}^{''} = (\pi(i) M_{ij} - i N_{ij}) t_i$;
Return
$\sigma = \{\sigma^{'}, \{S_i, S_i^{'}, S_i^{''}, S_i^{'''}\}_{i\in[1,l]}, \{P_j, P_{ij}^{'}, P_{ij}^{''}\}_{i\in[1,l], j\in[1,n]}\}$,
$C = (CT, \tau, \sigma)$

Decrypt: U wants to access the message stored by the cloud. U randomly selects $t \in Z_p^*$, then computes κ_{base}^t, T_i^t. To verify the user's identity, the cloud excutes Algorithm 9 to check whether $e(T_i^t, \Gamma_{ij} \Omega_{ij}^i) = e(\kappa_{base}^t, f_j)$. If true, CS sends C to CUA. Then CUA checks the validity of the time stamp τ by running Algorithm 10. If the current time is in τ, then CUA checks whether

$$\prod_{i\in[1,l]} e(S_{\pi(i)}, \Gamma_{ij} \Omega_{ij}^{\pi(i)})^{M_{ij}} = e(Y, f_j) e(g_2 g^{\mu}, P_j^{'}).$$

If it is true, then CUA obtains $C^{'}$ and sends it to U in a secret channel. Finally, U uses the retrieving key RK to compute

$$m = C_1 C^{'} e(g, S)^e.$$

Otherwise, U cannot restore the message m.

Algorithm 9: Decrypt 1

Input: κ_{base}^t, T_i^t
Output: C or $\perp$
Select $t \in Z_p^*$ and compute κ_{base}^t, T_i^t,
$S_{\pi(i)} = Q_{\pi(i)} T_{\pi_i}^{S_i^{'}} g_2^{S_i^{''}} g^{S_i^{'''}} = (T_{\pi(i)}^{v_{\pi(i)}})^{t_0} (g_2 g^{\lambda})^{t_{\pi(i)}} T_{\pi(i)}^{S_i^{'}} g_2^{S_i^{''}} g^{S_i^{'''}}$
$= T_{\pi(i)}^{v_i t_0} g_2^{t_i} g^{\mu t_i} = T^{\frac{v_i t_0}{\alpha + \gamma \pi(i)}} g_2^{t_i} g^{\mu t_i}$,
$P_j^{'} = P_j \prod_{i \in [i,l]} \Gamma_{ij}^{P_{ij}^{'}} \Omega_{ij}^{P_{ij}^{''}} = \prod_{i \in [1,l]} (f_j^{\alpha + \pi(i)\gamma})^{M_{ij} t_i}$;
if $e(T_i^t, \Gamma_{ij} \Omega_{ij}^i) = e(\kappa_{base}^t, f_j)$;
return C
else return $\perp$

Algorithm10: Decrypt 2

Input: $C, TK_{ue}, \Gamma_j, \Omega_j$
Output: $C^{'}$ or $\perp$
Check the validity of the time stamp τ;
if the current time is in τ;
if
$\prod_{i \in [1,l]} e(S_{\pi(i)}, \Gamma_{ij} \Omega_{ij}^{\pi(i)})^{M_{ij}} = e(Y, f_j) e(g_2 g^{\mu}, P_j^{'}), j \in [1, n]$
choose $w_{ji} \in Z_p$ such that $\prod_{j=1}^{n} \prod_{i=1}^{l_j} w_{ji} \lambda_{ji} = s$;
calculate $C^{'} = \frac{\prod_{j=1}^{n} \prod_{j=1}^{n} \prod_{i=1}^{l_j} (e(C_{ji}, p_j) e(D_{ji}, F_{\rho_j(i)}))^{w_{ji}}}{e(k_0, S) \prod_{j=1}^{n} e(k_j, S)}$
$= \frac{\prod_{j=1}^{n} \prod_{j=1}^{n} \prod_{i=1}^{l_j} (e(g^{c_j \lambda_{ji}} Y_{\rho_j(i)}^{-r_{ji}}, g^{w_{uj}}) e(g^{r_{jl_j}}, g^{y_{\rho_j} w_{uj}}))^{w_{ji}}}{e(g^{\beta_0} g^e, g^s) \prod_{j=1}^{n} e(g^{\beta_j} g^{c_j w_{uj}}, g^s)}$
$= \frac{e(g,g)^{s \sum_{j=1}^{n} c_j w_{uj}}}{e(g^{\beta_0} g^e, g^s) \prod_{j=1}^{n} e(g^{\beta_j} g^{c_j w_{uj}}, g^s)} = \frac{1}{e(g,g)^{c_0 s} e(g,g)^{es}}$;
return $C^{'}$
else return $\perp$
else return $\perp$

VII. Security Analysis

Theorem 1: If $q-PBDHE$ assumption holds with $(T^{'}, \varepsilon^{'})$, in the selective access structure model, our scheme is (T, q, ε) secure , where $T^{'} = T + O(T)$, $\varepsilon^{'} = \frac{1}{2}\varepsilon$.

Proof: There is a detailed proof in the Appendix.

Theorem 2: If the basic scheme [25] is secure, then the outsourcing solution is secure.

Proof: If our outsourcing scheme is broken by the adversary $\mathscr{A}$ with a non-negligible advantage, we can find an algorithm $\mathscr{B}$ to break the basic scheme with a non-negligible advantage. Let $\mathscr{C}$ be the challenger to the basic scheme $\mathscr{B}$. In the following steps, $\mathscr{B}$ interacts with $\mathscr{A}$.

Init: $\mathscr{A}$ sends the challenge access policy $\mathbb{A}^*$ to $\mathscr{B}$, and $\mathscr{B}$ sends $\mathbb{A}^*$ to $\mathscr{C}$ and gets the public parameters

$$PP = \{g, g_2, \{f_j\}_{j \in [1,n]}, p, G_0, G_1, g^{\nu}, C_0, H, H_1, H_2\}.$$

and authorities' public parameters

$$PK_j = \{C_j, (Y_{ij})_{a_{ij} \in A_j}\}$$

Setup: $\mathscr{B}$ sends PK, PK^* to $\mathscr{A}$.

Query phase 1: An empty table $\digamma$ and an empty set $\mathbb{K}$ are initialized by $\mathscr{B}$, $\mathscr{A}$ makes the private key queries as follows.

(1). $\mathscr{B}$ generates the private key SK for attribute set S. Then, $\mathscr{B}$ sets $\mathbb{K} = \mathbb{K} \cup \{S\}$. SK is sent to $\mathscr{A}$.

(2). $\mathscr{B}$ scans the tuple$\{S, SK, TK, RK\}$ in table $\digamma$. If there is such a tuple, transformation key TK is sent to $\mathscr{A}$. Otherwise, $\mathscr{B}$ randomly picks exponent $e_0 \in Z_p$ and computes $k_0 = g^{\beta_0} g^{e_0}$, then $\mathscr{B}$ stores tuple $\{S, k_0, (k_j, p_j, (Y_{ij})_{i \in [1,l_j]})_{j \in [1,n]}\}$. It's worth noting that $\mathscr{B}$ does not know the actual retrieving key $RK = e_0$.

Challenge: $\mathscr{A}$ selects two equal-length messages $\mathcal{M}_0, \mathcal{M}_1$ which have the same length and challenge access policy $\mathbb{A}^*$, then $\mathscr{B}$ sends $\mathcal{M}_0, \mathcal{M}_1, \mathbb{A}^*$ to $\mathscr{C}$. $\mathscr{B}$ gets the challenge ciphertext CT^* and sends it to $\mathscr{A}$.

Query phase 2: $\mathscr{A}$ performs private keys query and $\mathscr{B}$ makes the corresponding query like *phase 1*.

Guess: $\mathscr{A}$ and $\mathscr{B}$ make guesses.

If the guess μ of adversary $\mathscr{A}$ of our outsourcing scheme is correct, then the guess of $\mathscr{B}$ of basic scheme is also right. So we have a conclusion that if adversary $\mathscr{A}$ could attack our scheme at non-negligible advantage, algorithm $\mathscr{B}$ could attack basic scheme with an non-negligible advantage.

Theorem 3: If the CDH assumption holds, the proposed online-offline signature is unforgeable under chosen message attack.

Proof: The unforgeability of the scheme is briefly illustrated by the following proof.

Init: An adversary $\mathcal{A}$ chooses the challenge access structure $\mathbb{A}$, and $\mathcal{D}$ as challenger gets $\mathbb{A}$ from $\mathcal{A}$.

Setup: $\mathcal{D}$ picks two bilinear groups G_0, G_1 of prime order p generated by a generator g, and selects

$$g, g_2, f_1, f_2, \ldots, f_n \in G_0, e : G_0 \times G_0 \to G_1$$

Then, it outputs public parameters

$$PP = \{p, g, f_1, f_2, \ldots, f_n\}$$

$\mathcal{D}$ chooses $\alpha, \gamma \in Z_p$, and obtains the private key $SSK = (\alpha, \gamma)$ and the public key $SPK = \{\Gamma_{ij}, \Omega_{ij} | j \in [1, n]\}$. Finally, $\mathcal{D}$ sends GP, SPK to adversary $\mathcal{A}$, and keeps SSK secretly.

Signing key queries: The attribute set that does not satisfy the challenge structure is selected by the adversary $\mathcal{A}$. $\mathcal{A}$ posses the identity GID and its attribute set is u_0^*. The adversary $\mathcal{A}$ then asks the challenger $\mathcal{D}$ for the signing secret keys. If the identity GID is already registered with the CA, $\mathcal{D}$ returns the corresponding signing secret keys SSK. Otherwise, $\mathcal{D}$ selects α, γ randomly. Then, it sends the signing secret key to $\mathcal{A}$.

Offline signature queries: The ciphertext χ is adaptively selected by the adversary $\mathcal{A}$. Then, $\mathcal{A}$ sends it to the challenger $\mathcal{D}$ and makes some queries about the signature. The challenger $\mathcal{D}$ performs the offline signature algorithm$(M, SPK) \to \sigma^{'}$.

Online signature queries: The challenger runs the online signature algorithm $(SPK, \chi, \sigma^{'}) \to \sigma$ to generate the signature σ for the adversary $\mathcal{A}$.

Forgery: The adversary $\mathcal{A}$ generates the ciphertext χ^* and the forged signature σ. If the adversary $\mathcal{A}$ forges a signature

in a polynomial time with an non-neligible probability, $\mathcal{A}$ will solve the CDH assumption in a polynomial time with an non-neligible probability. The analysis is similar with that in [36]. We omit it here.

In the next section, we will demonstrate the security of the scheme in terms of access control and authentication. In the process of access control, no illegal user or AA can get the private key to decrypt the ciphertext. Only legitimate users can recover the plaintext with all the private keys. During anonymous authentication, any illegal user cannot be successfully authenticated. In addition, replay attack and user's privacy are also considered.

Theorem 4: Our scheme is collusion resisting among users or AAs during access control.

Proof: Preventing users' collusion is first considered. Suppose malicious users U_{adv}, U_{adv1} and U_{adv2} try to simulate the attacked target user U_{Target} when collusion happens. Without loss of generality, there are the following two cases:

$(\widehat{U}_{adv1} \cup \widehat{U}_{adv2}) \cap \widehat{AA_j} = \widehat{U}_{Target} \cap \widehat{AA_j}$ and $\widehat{U}_{Target} \cap \widehat{AA_j} = \widehat{U}_{adv} \cap \widehat{AA_j}$, where $\widehat{U}_{Target}$ denotes the attribute set of U_{Target} and $\widehat{U}_{adv}, \widehat{U}_{adv1}, \widehat{U}_{adv2}$ denote those of $U_{adv}, U_{adv1}, U_{adv2}$ respectively.

For two users U_{Target}, U_{adv} such that $\widehat{U}_{Target} \cap \widehat{AA_j} = \widehat{U}_{adv} \cap \widehat{AA_j}$, CA assembles all $e(g,g)^{\beta_j^{'}s}$ from related AAs and computes private key $\beta_j^{'}$. U_{adv} does not obtain the same private key as that of the user U_{Target} from AA_j, since AA_j generates different keys by $\beta_{j,Target}^{'} = PRF_{\beta_j}(U_{adv})$ and $\beta_{j,adv}^{'} = PRF_{\beta_j}(U_{adv})$ respectively. From what has been discussed above, U_{adv} does not obtain $\beta_{j,adv}^{'} = \beta_{j,Target}^{'}$ to compute the private key for decrypting related ciphertext blinded by $e(g,g)^{\beta_{j,Target}^{'}s}$. Similar to the above analysis, U_{adv1} and U_{adv2} do not make collusion because $(\widehat{U}_{adv1} \cup \widehat{U}_{adv2}) \cap \widehat{AA_j} = \widehat{U}_{Target} \cap \widehat{AA_j}$,.

We next demonstrate that two or more AAs can not collude. In our scheme, the protection of $H_1(u \parallel R_{uj})$ and privacy of attributes ensure AAs does not know the user's identity. R_{uj} is a different random value when facing different AA_j. AAs do not gather all relevant decryption keys by using the same pseudonym to realize collusion.

Suppose *n* attribute authorities collude and get the user's private key. But these private keys are only part of the user's decryption key. To decrypt the ciphertext, the key generated by CA is also obtained. However, *n* attribute authorities do not know the identity of the user, which means that they cannot get the private key generated by CA. The malicious centers cannot decrypt the ciphertext to get the plaintext. So our scheme resists AAs collusion attack.

Theorem 5: Our scheme resists collusion and replay attacks and protects the user's privacy during anonymous authentication.

Proof: Noting that legal users who have corresponding attributes can satisfy $e(T_i^t, (\Gamma_{ij}\Omega_{ij}^i)) = e(\kappa_{base}^t, f_j)$. Any two illegal users can not collude to access the sharing data. Let x_A and x_B be attributes of two users U_A and U_B, respectively. They have T_{x_A} and T_{x_B}, respectively. A value of $T_{x_B} = \kappa_{base}^{\frac{1}{a+bx_B}}$ can not be calculated by U_A, which means that illegal users can not combine to pass verification. So, our scheme resists collusion during authentication.

The proposed scheme also resists replay attacks. If the user is revoked, data cannot be replaced with outdated information. The user has to append a new time stamp τ and signs on the ciphertext. Because there is no τ, the user cannot have a valid signature and cannot authenticate successfully.

Finally, our scheme protects privacy of the user. When a user who possesses GID authenticates, T_i^t is uploaded and t is used to hide user's attributes exposed in T_i. Therefore, the user's identity information is hidden.

VIII. Performance analysis

A. Security and Functionality

We analyze the functionality and security of this scheme by comparing it to other existing schemes. [21], [25], [26], and [31] all involve MA-ABE in TABLE V. TABLE VI compares [21], [23], [32], and [33], where these schemes are based on ABS. $\surd$ indicates that the scheme is capable of this function, and $\times$ represents the opposite. TABLE V and TABLE VI show some features that our solution supports, such as ciphertext unforgeability, user's identity and attribute privacy, collusion resistance, public verifiability computation outsourcing, and anonymous authentication.

TABLE V
Security and function comparison of MA-ABE based schemes

Schemes	[21]	[25]	[26]	[31]	Ours
Ciphertext Unforgeability	×	×	√	×	√
User's identity privacy	×	×	√	√	√
User's attributes privacy	×	×	√	√	√
Collusion Resistance	√	√	√	×	√
Public Verifiability	×	√	×	×	√
Computation Outsourcing	×	√	×	×	√

TABLE VI
Security and function comparison of ABS based schemes

Schemes	[21]	[23]	[32]	[33]	Ours
Ciphertext Unforgeability	×	√	√	√	√
User's identity privacy	√	×	√	×	√
User's attributes privacy	×	×	×	×	√
Collusion Resistance	√	√	√	√	√
Offline-online	×	×	×	×	√
Anonymous authentication	√	√	×	√	√

B. Performance

In this subsection, we will compare the proposed scheme with others in terms of the computation cost of signature, verification, and decryption(user side), where the row of the access matrix is represented by l and the column by r. E, P and M represent the exponential, bilinear pairing, and multiplication operations.

TABLE VII
COMPARISON ON COMPUTATION COST

Schemes	Sign	Verify	Decryption(User side)
[13]	—	$(2l+3)E+3P+(2l+4)M$	$lE+(l+2)P+2lM$
[21]	$(lN_A+2N_A)M$+$(2lN_A+3N_A+2)E$	$2lE+(l+2N_A)P$	$(5l+1)E+4lP+(l+1)M$
[23]	$(2l+lN_A+4)E+(lN_A+2)M$	$(lN_A+N_A+1)P+(2lN_A+N_A)E+(lN_A+N_A)M$	$(l+1)P+(l+2)M+E$
Ours	$(2N_A+6)lM$	$(l+N_A)E$	$E+P+M$

TABLE VII compares computational overhead. The calculation cost of [13], [21] and [23] increases linearly with l and N_A. Our scheme places much computation in the offline phase. Therefore, online signature operations are greatly reduced. In the certification phase, the computation cost of [13], [21] and [23] is linearly related to l or N_A. But in our scheme, CUA is in charge of verification and the user hides its attributes in the authentication, so the amount of calculation of the user only needs some exponential operations. Similarly, our decryption calculation only needs $E+P+M$ because of the outsourcing of calculation. As the number of users continues to increase, our scheme still has a computational advantage over the others. Based on the above analysis, our scheme lowers computation overhead in signing, verification, and user decryption.

TABLE VIII shows a comparison of storage cost. Let $|G|$ and $|G_1|$ represent the length of elements in groups G and G_1, respectively. S is the number of the user's attributes and U_d is the number of attributes when decrypting. n is the number of attributes per attribute authority.

Because the encryption scheme requires an LSSS policy, the ciphertext and decryption of these schemes are all linearly related to l. The storage cost of the proposed scheme is $(4l+N_A+2lN_A)|G|+(2lN_A+N_A+S+2)|G_1|$, and decryption cost is $2|G|+(S+2N_A+nN_A+1)|G_1|$. However, CS has much storage capacity, and we only consider the storage overhead of users. The storage of ciphertext in [21], [25] and [26] increases linearly with l. We set up a simulation experiment based on the PBC library and set a subset of the policy attribute set as the attribute set. The attributes meet the access policy. We use five attribute authorities during the experiment and set 20 attributes for each AA.

In the signing phase, as shown in Fig. 4, the computation overhead of [21], [23] and our scheme is linearly related to the number of policy attributes l. [21] executes the operations which include $10l+17$ exponentiations and $5l+10$ multiplications. [23] includes $5l+2$ multiplication operations and $7l+4$ exponentiation operations. The calculation cost of ours is lower than them.

Fig. 5 shows the computation cost of verification. [13] includes $2l+3$ exponentiation operations, $2l+4$ multiplication operations, and 3 pairing operations. $2l$ exponentiation operations and $10+l$ pairing operations are needed in [21]. [23] includes $10l+5$ exponentiation operations, $5l+5$ multiplication operations, and $5l+6$ pairing operations, and our scheme only needs $l+5$ exponentiation operations. Fig. 6, which displays the computational cost of decryption on the user side, shows that the cost of our scheme is less than others.

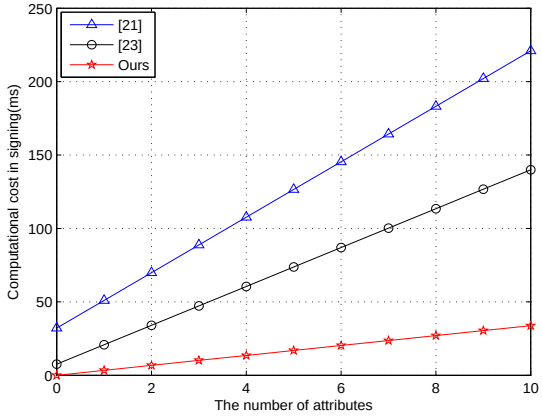

Fig. 4. Computational time of signing algorithm

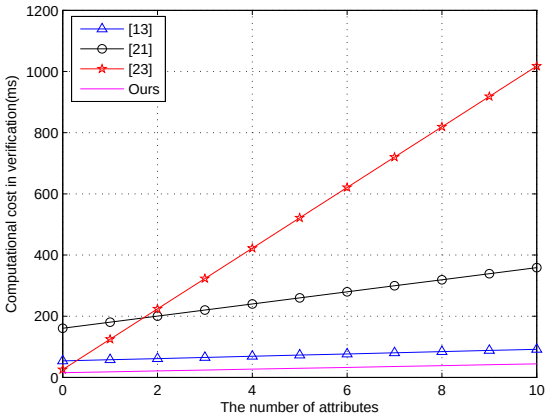

Fig. 5. Computational time of verification algorithm

IX. CONCLUSION

We proposed a secure sharing framework based on multi-authority attribute-based encryption for PHRs system. In this scheme, the identity and attributes of the user are hidden and known only to the trusted central authority. To prevent cloud server from tampering with ciphertext or spoofing end users, an anonymous authentication based on attribute-based signature is proposed. In the whole access-control process, only authorized users can access and obtain messages. For achieving lightweight computation, online and offline technique and outsourcing operations are used. Compared with the existing works, the proposed scheme not only keeps the encrypted PHRs to resist collusion attacks and not to be

TABLE VIII
COMPARISON ON STORAGE COST

Schemes	Public key	Ciphertext	Decryption
[21]	$(3N_A+1)\|G_1\|$	$(3l+1+2nN_A)\|G_1\|$	$\|U_d\|\|G_1\|$
[25]	$(3+nN_A+N_A)\|G_1\|$	$(2lN_A+2)\|G_1\|$	$(2N_A+nN_A+1)\|G_1\|$
[26]	$(5N_A+3nN_A)\|G_1\|$	$(3N_A+2l+1)\|G_1\|$	$(6N_A+\|U_d\|)\|G_1\|$
Ours	$(2+3N_A+nN_A)\|G_1\|$	$(2lN_A+N_A+S+2)\|G_1\|$ $+(4l+N_A+2lN_A)\|G\|$	$2\|G\|+(S+2N_A$ $+nN_A+1)\|G_1\|$

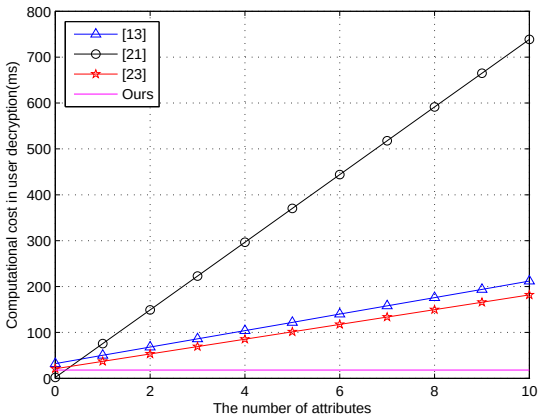

Fig. 6. Computational time of decryption algorithm

forged during the period of sharing, but also achieves privacy preserving, which enhances patients' control to their PHRs. To meet the higher security and efficiency of practical application scenarios, this solution can be extended from the following two aspects.

Anonymous authentication between user and authority: Because authorities were assumed to be dishonest, authentication between user and authority is worth investigating.

Traceability: Considering the practicality of the scheme, when an illegal user fails to authenticate its identity, the system obtains the current user's identity through traceable technology.

ACKNOWLEDGMENT

We thank LetPub (www.letpub.com) for its linguistic assistance during the preparation of this manuscript. Furthermore, the authors would like to thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] L. Tbraimi, M. Asim, M. Petkovi, "Secure management of personal health records by applying attribute-based encryption, In Proceeding of the International Workshop on Wearable Micro and Nano Technologies for Personalized Health(pHealth)," in *Oslo, Norway*, Jun.2009, pp.71–74.

[2] J. Akinyele, M. Pagano, M. D. Green, "Securing electronic medical records using attribute-based encryption on mobile devices," in *Proceeding of the ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, Oct.2011, pp.75–86.

[3] S. Narayan, M. *Gagné*, R. Safavi-Naini, "Privacy preserving EHR system using attribute-based infrastructure," in *proceeding of the ACM Cloud Computing Security Workshop, Chicago*, Oct.2010, pp.47–52.

[4] J. Lai, R. H. Deng, Y. Li, "Fully secure ciphertext-policy hiding CP-ABE," in *Proceedings of the International Conference on Information Security Practice and Experience*, Jun.2011, pp.24–39.

[5] J. Sun, Y. Fang, "Cross-domain data sharing in distributed electronic health record systems," in *IEEE Trans.Parallel Distrib.Syst.*, Jun.2009, pp.754–764.

[6] M. Li, S. Yu, Y. Zheng, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," in *IEEE Trans.Parallel Distrib.Syst.*, 2013, pp.131–143.

[7] X. Liang, M. Barua, R. Lu, "HealthShare: Achieving secure and privacy-preserving health information sharing through health social networks," in *Comput.Commun.*, 2012, pp.1910–1920.

[8] R. Lu, X. Lin, X. Shen, "SPOC: A secure and privacy-preserving oppotunistic computing framework for mobile-healthcare emergency," in *IEEE Trans.Parallel Distrib.Syst.*, 2013, pp.614–624.

[9] X. Zhou, J. Liu, Q. Wu, "Privacy preservation for outsourced medical data with flexible access control," in *IEEE Access.*, Jun.2018, pp.14827–14841.

[10] S. Jiang, X. Zhu, and L. Wang, "EPPS:Efficient and privacy-preserving personal health information sharing in mobile healthcare social networks," in *Sensors.*, 2015, pp.22419–22438.

[11] K. Yang, Q. Han, and H. Li, "An efficient and fine-grained big data access control scheme with privacy-peserving policy," in *IEEE Internet Things.*, 2017, pp.563–571.

[12] M. Yang, T. Zhang, "Efficient privacy-preserving access control scheme in electronic health records system," in *Sensors.*, 2018, pp.3520–3525.

[13] Y. Liu, Y. Zhang, and J. Ling, "Secure and fine-grained access control on e-healthcare records in mobile cloud computing," in *Future Gener.Comp.Sy.*, 2018, pp.1020–1026.

[14] A. Sahai, B. Waters, "Fuzzy Identity-Based Encryption," in *Proc.EUROCRYPT*,vol.LNCS 3494, May.2005, pp.457–473.

[15] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc.13th ACM conference on Computer and Communication Security*, 2006, pp.457–473.

[16] J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc.IEEE Symposium on Security and Privacy*, 2007, pp.321–334.

[17] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography Conference*, Springer, 2007, pp.515–534.

[18] J. Wei, W. Liu and X. Hu, "Seure and efficient attribute-based access control for multiauthority cloud storage," in *IEEE SYSTEM JOURNAL*, June.2018, pp.1731–1742.

[19] H. K. Maji, M. Prabhakaran and M. Rosulek, " Attribute-Based Signatures," in *Topics in Cryptology - CT-RSA*, vol.6558, 2011, pp.376–392.

[20] S. Zhang, P. Chen and J. Wang, "Online/Offline Attribute Based Signature," in *Ninth International Conference on Broadband and Wireless Computing, Communication and Applications.IEEE*, 2014, pp.566–571.

[21] S. Ruj, M. Stojmenovic, A Nayak. "Decentralized access control with anonymous authentication of data stored in cloud," in *IEEE Transactions on Paralell and Distributed Systems*, Feb.2014, pp.384–394.

[22] X. Li, J. Jiang and Y. Chen, "Fully decentralized authentication and revocation scheme in data sharing systems," in *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications*, 2018, pp.680–686.

[23] X. Liu, Y. Xia, S. Jiang, F. Xia and Y. Wang "Hierarchical attribute-based access control with authentication for outsourced data in cloud computing," in *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*,2013, pp.477–484.

[24] H. Nicanfar, P. Jokar, K. Beznosov and V. C. M. Leung, "Efficient authentication and key management mechanisms for smart grid communications," in *IEEE SYSTEMS JOURNAL*, Jun.2014, pp.629–640.

[25] S. Hu, J. Li and Y. Zhang, "Improving security and privacy-preserving in multi-Authorities ciphertext-policy attribute-based encryption," in

KSII Transactions on Internet and Information Systems, vol.12,NO.10, Oct.2018, pp.5100–5119.
[26] J. Han, W. Susilo, Y. Mu, J. Zhou and M. Au, "Improving privacy and security in decentralized ciphertext-policy attribute-based encryption," in *IEEE Transactions on Information Forensics and Security*,vol.10, Mar.2015.
[27] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *PKC*, 2014, pp.293–310.
[28] A. Shamir, "Identity-based cryptosystems and signature schemes," in *CRYPTO84*, 1984, pp.47–53.
[29] S. Guo, Y. Zeng, "Attribute-based signature scheme," in *ISA08*, 2008, pp.509–511.
[30] R. Bobba, O. Fatemieh, F. Khan, C. A. Gunter, and H. Khurana, "Using attribute-based access control to enable attribute-based messaging," in *IEEE Computer Society*, 2006, pp.403–413.
[31] J. Xu, Q. Wen, W. Li, J. Shen and D. He, "Succinct multi-authority attribute-based access control for circuits with authenticated outsourcing," in *Soft Comput.*, 2017, pp.5265–5279.
[32] Y. S. Rao, "A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing," in *FutureGener.Comput.Syst.*, vol.67, 2017, pp.133–151.
[33] G. Yu, Z. Cao, "Attribute-based signcryption with hybrid access policy," in *Peer-to-Peer Netw.Appl.*, vol.10, 2017, pp.253–261.
[34] T. Okamoto, K. Takashima, "Decentralized arrtibute-based signature," in *Proc.Int.Workshop Public Key Cryptor.*, 2013, pp.125–142.
[35] X. Yan, H. Ni, Y. Liu and D. Han, "Privacy-preserving multi-authority attribute-based encryption with dynamic policy updating in PHR," in *Computer Science and Information Systems.*, 2019, pp.831-847.
[36] D. Boneh, B. Lynn and H. Shacham, "Short signatures from the weil pairing," in *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, 2001, pp.514-532.
[37] D. Li, J. Liu, Q. Wu and Z. Guan, "Efficient CCA2 secure flexible and publicly-verifiable fine-grained access control in fog computing," in *IEEE Access.*, 2019, pp.11688-11697.
[38] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai and R. Attia, "Accountable privacy preserving attribute based framework for authenticated encrypted access in clouds," in *J. Parallel Distrib. Comput.*135 (2020) 1-20.

Leyou Zhang received the M.S. and Ph.D. degrees from Xidian University, in 2002 and 2009, respectively. He is currently a Professor with Xidian University. His current research interests include cryptography, network security, cloud security, and computer security.

Yadi Ye received the B.S. degree in mathematics from Shandong Jianzhu University, in China, in 2018. She is currently pursuing the M.S. degree in applied mathematics with Xidian University, China. Her current interests include cryptography and cloud security.

Yi Mu (SM'00) received the Ph.D. degree from the Australian National University, Canberra, ACT, Australia, in 1994. In 2018, he was a Professor of computer science with the University of Wollongong, Wollongong, NSW, Australia. He is currently a Professor with the College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China. His current research interests include blockchain, cybersecurity, and cryptography. Prof. Mu was the Editor-in-Chief of the International Journal of Applied Cryptography and has served as an Associate Editor for several other international journals.

X. Appendix

Theorem 1: If $q-PBDHE$ assumption holds with $(T^{'}, \varepsilon^{'})$, our scheme is (T, q, ε) secure in the selective access structure model, where $T^{'} = T + O(T)$, $\varepsilon^{'} = \frac{1}{2}\varepsilon$.

Proof: If a (T, q, ε) adversary can break our scheme, then we show there is an algorithm $\mathcal{B}$ can break the decisional $q-PBDHE$ assumption. The challenger selects the bilinear group (e, p, G_0, G_T) and a generator $g \in G$. Let $\vec{Y} = g, g^s, g^a, \cdots, g^{a^q}, g^{a^{q+2}}, \cdots, g^{a^{2q}}, g^{s\cdot b_j}, g^{\frac{a}{b_j}}, \cdots, g^{\frac{a^q}{b_j}}, g^{\frac{a^{q+2}}{b_j}}, \cdots, g^{\frac{a^{2q}\cdot b_k}{b_j}}, g^{\frac{a\cdot s\cdot b_k}{b_j}}, g^{\frac{a^2\cdot s\cdot b_k}{b_j}} \cdots, g^{\frac{a^q\cdot s\cdot b_k}{b_j}}$ $(\forall 1 \le j, k \le q, k \ne j)$. If R is selected in G_T at random. The challenger outputs $v \in \{0, 1\}$. If $v = 1$, then $(\vec{Y}, \Omega = R)$ is sent to $\mathcal{B}$. Otherwise, $\vec{Y}, \mathcal{B}$ outputs a guess $v^{'}$ on v and $\Omega = e(g, g)^{a^{q+1}s}$ is sent to $\mathcal{B}$.

Setup: Assume that I^* is an A_j's index set , $\mathcal{A}$ submits access structure $\mathbb{A} = \{\widehat{M_j^*}, \rho_j^*\}_{j\in I^*}$. If $\widetilde{A^*}$ specifies $(\widehat{M^*}, \rho_j^*)_{j\in I^*}$ and the access structure does not satisfy attributes which are chosen by $\mathcal{A}$ to make private keys query. $\mathcal{B}$ chooses 3 hash functions: $H : \{0,1\}^* \to G_0, H_1 : \{0,1\}^* \times G_0 \to \{0,1\}^*, H_2 : G_0 \times \{0,1\}^* \times \{0,1\}^* \to G_0$. Pick random group elements $g, g_2, h_1, \cdots, h_n \in G_0$. CA randomly chooses exponents $c_0, v \in Z_p$, and generates two public -secret key pairs $(g^v, v), (C_0, c_0)$, where $C_0 = e(g, g)^{c_0}$. The public parameters are

$$PP = \{g, g_2, \{f_j\}_{j\in[1,n]}, P, G_0, G_1, g^v, C_0, H, H_1, H_2\}.$$

Authorities Setup: If the authority $AA_j \ne AA^*$, $\mathcal{B}$ gives the following simulation. AA_j selects $\beta_j, c_j \in Z_p$ at random, and publishes public key $PK_j = \{C_j, (Y_{ij})_{a_{ij}\in AA_j}\}$, private keys $SK_j = \{\beta_j, c_j, (y_{ij})_{a_{ij}\in AA_j}\}$.

For the authority AA^*, $\mathcal{B}$ randomly picks $\alpha_0, \beta \in Z_p$ and computes $B = g^\beta$. For the attribute $a_x \in AA^*$, $\widehat{X}$ represents the set of indices i, namely $\rho(i)^* = a_x$.

(1). If attribute $a_x \in \hat{A}^*$ and $a_x = \rho(i)^*$, then $\mathcal{B}$ randomly selects $y_x \in Z_p$. $\mathcal{B}$ computes

$$Y_x = g^{y_x} \prod g^{\frac{a\widehat{M_{i,1}^*}}{b_i}} g^{\frac{a^2\widehat{M_{i,2}^*}}{b_i}} \cdots g^{\frac{a^{n^*}\widehat{M_{i,n^*}^*}}{b_i}}.$$

(2). If attribute $a_x \in \widehat{A^*}$ and $a_x \ne \rho(i)^*$, then $\mathcal{B}$ picks $z_x \in Z_p$ at random and computes $Y_x = g^{y_x}$. $\mathcal{B}$ sends the authority $\widetilde{A^*}$'s public parameter $PK^* = (B, Y_x)$ to $\mathcal{A}$.

User Register: Assume that $\mathcal{A}$ represents some legal users, such as U. The CA generates the anonymous identity credential

$$AID = \{Y_{ij}, Pid_{uj}, \Lambda_{AA_j}, Sig_{a_{ij}}\}_{a_{ij}\in\hat{U}, j\in[1,n]}$$

to $\mathcal{A}$.

Phase 1: Performing secret key query as follows.

Making secret key query for U with pseudonyms $\{R_{uj}\}_{\widehat{U}\cap A_j\neq\emptyset}$. It is worth noting that the corresponding $\widehat{U}$ does not satisfy $\widehat{M^*}$. $\mathcal{B}$ calculates if

$$e(Sig_{a_{ij}},g)=e(H_2(Y_{ij},Pid_{uj},\Lambda_{AA_j}),g^v).$$

(1) For $\widehat{AA_j}=\widehat{AA^*}$, $\mathcal{B}$ randomly selects $\beta_0\in Z_p,\beta_0^{'}=PRF_{\beta_0}(R_{uj})$, $\mathcal{B}$ sets $\beta=\beta_0^{'}+a^{q+1}$ and computes $Y^*=e(g,g)^{\beta}=e(g^a,g^{a^q})e(g,g)^{\beta_0^{'}}$.

(a). If $a_x\in AA^*\cap\widehat{U}$, and $\rho(i)^*=a_x$, $\mathcal{B}$ randomly selects $r\in Z_p,\vec{V}=(v_1,v_2,\cdots,v_{n^*})\in Z_p^*$ such that $v_1=-1$ and $\widehat{M^*}\cdot\vec{V}=0$ for all $\rho(i)^*\in\widehat{AA^*}\cap\widehat{U}$. $\mathcal{B}$ computes $k=g^{\beta_0^{'}}g^{ra}\prod_{i=2}^{n^*}g^{v_ia^{q-i+2}}$ and $P=g^r\prod_{i=1}^{n^*}(g^{a^{q-i+1}})^{v_i}=g^w$, $w=r+v_1a^q+\cdots+v_{n^*}a^{q-n^*+1}$. $\mathcal{B}$ computes $F_x=P^{y_x}\prod_{i\in\widehat{X}}\prod_{j=1}^{n^*}(g^{\frac{ra^j}{b_i}}\prod_{k=1,k\neq n^*}^{n^*}g^{\frac{v_ka^{q+1+j-k}}{b_i}})^{\widehat{M_{ij}^*}}=Y_x^w$.

(b). If $a_x\in\widehat{AA_j}\cap\widehat{U}$ and $\rho(i)^*\neq a_x$, $\mathcal{B}$ computes $Y_x=g^{y_x}$ and $F_x=P^{z_x}=g^{y_xw}=Y_x^w$.

For $a_x\in\widehat{AA^*}$ $k=g^{\beta_0}g^eg^{ra}\prod_{i=2}^{n^*}g^{v_ia^{q-i+2}}=g^{\beta}g^eg^{aw}$, $P=g^w$, the simulation of the secret keys is perfect. $\mathcal{B}$ sends the private key $\{k,P,(F_x)_{a_x\in\widehat{U}\cap\widehat{AA^*}}\}$ to $\mathcal{A}$.

(2). For $\widehat{AA_j}\neq\widehat{AA^*}$, $\mathcal{B}$ randomly selects $\beta_j,w_j\in Z_p$ and computes $\beta_j^{'}=PRF_{\beta_j}(R_{uj}),k_j=g^{\beta_j^{'}}g^{x_jw_j},P_j=g^{w_j},F_{ij}=Y_{ij}^{w_j}$. Finally, $\mathcal{B}$ obtains the private key $\{k_j,p_j,(Y_{ij})_{a_{ij}\in\widehat{U\cap AA_j}}\}$ from $\mathcal{A}$.

(3). For CA, $\mathcal{B}$ computes $\beta_0=c_0-\sum_{j=1}^n\beta_j^{'},k_0=g^{\beta_0}g^e$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ presents two equal-length messages M_0,M_1. $\mathcal{B}$ selects $\hat{\theta}$ by flipping a coin with $\{0,1\}$.

(1). For $\widehat{AA^*}$, $\mathcal{B}$ gets $X=g^a$. Then, $\mathcal{B}$ randomly selects $r_1,r_2,\cdots,r_{l^*},f_2,f_3,\cdots,f_{n^*}\in Z_p$, and sets $\vec{f}=(s,as+f_2,sa^2+f_3,\cdots,sa^{n^*-1}+f_{n^*})$, which is applied to share the secret s. $\mathcal{B}$ calculates

$$C_k=Z_{\rho(k)^*}^{r_k}(\prod_{j=2}^{n^*}g^{a\widehat{M_{ij}^*}f_j})g^{-z_{\rho(k)^*}b_ks}$$

$$(\prod_{l\in\Lambda_i}\prod_{j=1}^{n^*}g^{a^js\widehat{M_{kj}^*}(\frac{b_k}{b_l})}),$$

$$D_k=g^{-r_k}g^{-sb_k},$$

where $k=1,2,\cdots,l^*$.

(2). For $AA_j(j\in I^*,AA_j\neq AA^*)$, $\mathcal{B}$ computes $X_j=g^{x_j}$ and randomly chooses $r_{j1},r_{j2},\cdots,r_{jl_j},f_{j2},f_{j3},\cdots,f_{jn_j}\in Z_p$. $\mathcal{B}$ sets $\vec{f_j}=(s,f_{j2},\cdots,f_{jn_j})$ which is applied to share the secret s. $\mathcal{B}$ calculates $C_{jk}=g^{x_j\widehat{M_j^{k1}}}\prod_{i=2}^{n_j}g^{f_{ji}\widehat{M_j^{ki}}}Z_{\rho_j(k)}^{-r_{jk}},D_{jk}=g^{r_{jk}}(k=1,2,\cdots,l_j)$ and $C_0^*=M_{\hat{\theta}}e(g^{\alpha_0},g^s)\Omega e(g^{\alpha_o^{'}},g^s))\prod_{j\in I^*,AA_j\neq AA^*}e(g,g)^{\alpha_j^{'}s}$. That is $C_0^*=M_{\hat{\theta}}e(g,g)^{x_0s}$, $\mathcal{B}$ finally computes

$$CT^*=\{C_0^*,X,(C_k,D_k)_{k=1}^{l^*},(X_j,(C_{jk},D_{jk})_{k=1}^{l_j})_{j\in I^*}\}.$$

Phase 2: Continue making secret key queries like *Phase 1*.

Guess: After successfully simulating the game, $\mathcal{A}$ returns his/her guess $\tilde{\theta}$ of $\hat{\theta}$ flipped by $\mathcal{B}$, $\mathcal{B}$ outputs $v^{'}=0$, which shows $v=0$, $\Omega=e(g,g)^{a^{q+1}s}$ and CT^* is well formed about M_0. If $\tilde{\theta}$ is not equal to $\hat{\theta}$, then $\mathcal{B}$ outputs $v^{'}=1$, which shows that Ω is a random number in G_T, $v=1$. If $v=0$, $Pr[\mathcal{B}(\vec{Y},\Omega=e(g,g)^{a^{q+1}s})]>\frac{1}{2}+\varepsilon$. If $v=1$, $Pr[\tilde{\theta}\neq\hat{\theta}\mid v=1]=\frac{1}{2}$,which indicates $\mathcal{A}$ outputs $\tilde{\theta}\neq\hat{\theta}$ with negligible advantage.