A Novel Privacy-Preserving Decentralized Ciphertext-Policy Attribute-Based Encryption with Anonymous Key Generation

Hongjian Yin[1], Leyou Zhang[1(✉)], and Yi Mu[2]

[1] Xidian University, Xi'an 710126, Shaanxi, China
xidianyhj@163.com, lyzhang@mail.xidian.edu.cn
[2] University of Wollongong, Wollongong, NSW 2522, Australia
ymu@uow.edu.au

Abstract. A privacy-preserving decentralized ciphertext-policy attribute-based encryption (CP-ABE) scheme is a variant of the multi-authority attribute-based encryption schemes where it requires neither a central authority nor cooperation among authorities for issuing secret keys. It also featured the privacy-preserving and resisting user collusion. However, previous privacy-preserving decentralized CP-ABE schemes can only hide user's partial information, such as global identifier (GID), but user's attribute information leaked to the authority may be sensitive which will lead to privacy disclosure. To overcome this shortcoming, we propose an improved privacy-preserving decentralized CP-ABE scheme with anonymous key generation protocol, where it can prevent authorities from learning any information about user's both GID and attributes. Theoretical analysis and simulation results demonstrate that the proposed scheme is secure and efficient. In the standard model, its security is reduced to a standard decisional bilinear Diffie-Hellman complexity assumption.

Keywords: Privacy-preserving · Multi-authority
Decentralized CP-ABE · Anonymous key generation protocol

1 Introduction

Attribute-based encryption (ABE) is one of flexible public key encryption that allows for fine-grained access control on encrypted data. In an ABE scheme, the data owner can specify an access policy over a set of attributes, where these users whose attributes satisfy the policy can access the encrypted data. Since the first ABE scheme was proposed by Sahai and Waters [1], it has been intensively researched and further developed. There are two types of ABE schemes, which are called ciphertext-policy attribute-based encryption (CP-ABE) [2] and key-policy attribute-based encryption (KP-ABE) [3]. In a CP-ABE scheme, ciphertext is related to access structure and the secret keys of user are associated with

X. Sun et al. (Eds.): ICCCS 2018, LNCS 11065, pp. 435–446, 2018.
https://doi.org/10.1007/978-3-030-00012-7_40

an attribute set. Only the user whose secret keys satisfy the access structure associated with the ciphertext will be able to decrypt the ciphertext successfully. In contrast, in a KP-ABE scheme, ciphertext is related to an attribute set and the secret keys of user are associated with access structure. The user will be able to decrypt ciphertext only if the attributes associated with the ciphertext satisfy the access structure of the private key.

Most of ABE proposals are issued from the single authority. The single authority generates the private keys of the users and verifies all the attributes by itself. So it is impractical in some cases especially large scale attributes set. Chase further developed the ABE scheme and proposed the notion of multi-authority ABE [4]. Compared with previous ABE schemes, Chase's scheme supports multiple authorities to distribute attributes instead of a single authority. Specifically, her multi-authority ABE scheme allows any polynomial number of independent authorities to monitor attributes and distribute secret keys. In order to resist user collusion attacks, user secret keys have to be tied to a global identifier (GID) and a fully trusted central authority is necessary to issue a unique key to each user. However, this central authority has ability to decrypt every ciphertext in this system. The whole system will fail if the central authority is corrupted.

To solve the above problem, Lewko and Waters proposed a new multi-authority ABE system named decentralized ABE [5]. In their scheme, the central authority is removed and each independent authority can create public key and issue attribute secret keys to different users. A user can encrypt data in terms of any boolean formula over attributes issued from any chosen set of authorities. In addition, authorities are completely independent, such that every authority can join or leave the system without the necessity of reinitializing the system. And some corruption authorities will not affect the other uncorrupted authorities.

Some recent works about decentralized ABE have focused on achieving privacy preserving [6,7,9,10]. Specially, Han et al. further developed the decentralized ABE and proposed the privacy-preserving decentralized KP-ABE [6]. In their scheme, each authority can issue secret keys to a user independently without knowing anything about the user's GID. In addition, their scheme is tolerant against maximum $(N-1)$ authorities colluding. It means that the scheme is secure if the number of the corrupted authorities is not more than $(N-1)$, where N is the number of the authorities in the whole system. However, Ge et al. [11] pointed out that it did not resist user collusion attack. Subsequently, a modified privacy-preserving decentralized KP-ABE scheme was proposed by Rahulamathavan et al. [9]. Their scheme mitigates the user collusion attack employing anonymous key issuing protocol and achieves user's GID hidden.

In 2014, Han et al. proposed another type privacy-preserving decentralized ABE, named privacy-preserving decentralized CP-ABE [7]. User privacy protection is further considered in this scheme where both users' GID and attribute information are hidden from the authorities. It means that a user can get his/her attribute secret keys from multiple authorities without revealing any information about his/her GID and attributes. Unfortunately, this scheme is also vulnerable to collusion attack [12], which means that some unauthorized users whose

attributes do not satisfy the ciphertext policy combine their secret keys together and then decrypt the ciphertext successfully. Additionally, in this scheme, the authority can figure out the attributes information from key extract protocol by running the decisional Diffie-Hellman test (DDH-test) $e(\Theta_2, Z_x) \stackrel{?}{=} e(\Psi_x^2, g)$. Moreover, its security is reduced to q-strong Diffie-Hellman assumption which is a strong hardness assumption.

Until now, as described in [12], it is an open problem to construct a decentralized ABE scheme in which both GID and attributes are hidden to support privacy preserving. In this paper, a novel privacy-preserving decentralized ciphertext-policy attribute-based encryption is proposed to answer this open problem. In our scheme, both user's GID and attributes are hidden. Specifically, the proposed scheme can prevent authorities from learning any information about user's GID and attributes. Different authorities are able to issue secret key independently to users and need not even be aware of each other. The security of the proposed scheme is reduced to a standard decisional bilinear Diffie-Hellman complexity assumption. Furthermore, compared with some previously known multi-authority ABE schemes, our privacy-preserving decentralized CP-ABE scheme is efficient.

2 Preliminaries

2.1 Access Structure

Our construction will employ AND-gate on multi-valued attributes access structure, which is similar to what used in [15]. It is described as follows.

Let $\mathbb{U} = \{att_1, att_2, \cdots, att_n\}$ be a set of attributes. For $att_i \in \mathbb{U}$, $S_i = \{v_{i,1}, v_{i,2}, \cdots v_{i,m_i}\}$ is a set of possible values, where m_i is the number of possible values for each att_i. Let $L = [L_1, L_2, \cdots L_n]$ be an attribute list for a user where $L_i \in S_i$. Let $\mathbb{A} = [w_1, w_2, \cdots w_n]$ be an access structure where $w_i \in S_i$. The notation $L \models \mathbb{A}$ expresses that an attribute list L satisfies an access structure $\mathbb{A}$ and $\not\models$ refers to not satisfy symbol.

2.2 Commitment Scheme

A commitment scheme allows someone to commit a chosen value without leaking this value for a period of time and reveal the committed value later when it is needed. There are two properties in a commit scheme, *binding* and *hiding*. Binding: once the value has been committed to, its owner will not be able to change the value. Hiding: the value remains unreleased until its owner release it later. In our scheme, we will follow Pedersen's commitment scheme which is a perfect hiding commitment scheme introduced in [16], it is defined as follows.

- Setup: Let $\mathbb{G}$ be a group with prime order. $g_0, g_1, \cdots, g_l$ are the generators of group $\mathbb{G}$.
- Commit: This algorithm takes messages $(m_1, m_2, \cdots, m_l)$ and a random number $r \in_R \mathbb{Z}_p$ as input, returns the commitment $T = g_0^r \prod_{j=1}^{l} g_j^{m_j}$.

– Decommit: The algorithm decommits the commitment with the random value r. If the commitment is correct, it outputs 1, otherwise outputs 0.

2.3 Zero-Knowledge Proof

A zero-knowledge proof system is always run between prover and verifier. The prover wants to convince the verifier some knowledge is true, but without revealing the knowledge during the exchange. In our scheme, we will use the zero-knowledge proof scheme proposed by Camenisch and Stadler [17]. The scheme is defined as follows.

We denote a zero-knowledge proof of integers α, β and γ by $PoK\{(\alpha, \beta, \gamma) : y = g^{\alpha}h^{\beta} \wedge \tilde{y} = \tilde{g}^{\alpha}\tilde{h}^{\gamma}\}$, where g, h are the generators of group $\mathbb{G}$ and $\tilde{g}, \tilde{h}$ are the generators of group $\tilde{\mathbb{G}}$. The integers α, β and γ are the knowledge, while other values can be used to verify the equations by the verifier.

2.4 K-out-of-n Oblivious Transfer

A k-out-of-n oblivious transfer (denoted by OT_n^k) protocol involves two parties, the sender S and the receiver R. The sender S has n messages $m_1, m_2, \cdots, m_n$ and the receiver R wants to obtain some party of them $m_{\sigma_1}, m_{\sigma_2}, \cdots, m_{\sigma_j}$, where $j < n$. In doing this process, R only obtains the massages what he/she choices and S does not know which massages are chosen by R. In our construction, we will employ the efficient OT_n^k-II scheme which was proposed by Chu and Tzeng [18], the scheme is described in Algorithm 1. Let $\mathbb{G}_q$ be the subgroup of $\mathbb{Z}_p^*$ with prime order q, g be a generator of $\mathbb{G}_q$, and $p = 2q + 1$ is also prime. Let $H_1^* : \{0,1\}^* \to \mathbb{G}_q$, $H_2^* : \mathbb{G}_q \to \{0,1\}^l$ be two collision-resistant hash functions. Let messages be of l-bit length.

Algorithm 1. k-Out-of-n Oblivious Transfer

System parameters: $(g, H_1^*, H_2^*, \mathbb{G}_q)$;

1: R computes $w_{\sigma_j} = H_1^*(\sigma_j)$ and $A_j = w_{\sigma_j} g^{a_j}$, where $a_j \in_R \mathbb{Z}_q$ and $j = 1, 2, \cdots, k$
2: R sends $A_1, A_2, \cdots, A_k$ to S
3: S computes $y = g^x$, $D_j = (A_j)^x$, $w_i = H_1^*(i)$, and $c_i = m_i \oplus H_2^*(w_i^x)$, where $x \in_R \mathbb{Z}_q$, $i = 1, 2, \cdots, n$, and $j = 1, 2, \cdots, k$
4: R sends $y, D_1, D_2, \cdots, D_k, c_1, c_2, \cdots, c_n$ to S
5: R computes $K_j = D_j / y^{a_j}$ and gets $m_{\sigma_j} = c_{\sigma_j} \oplus H_2^*(K_j)$ for $j = 1, 2, \cdots, k$

3 Definition and Security Model

3.1 Definition of DCP-ABE

A definition of decentralized ciphertext-policy attribute-based encryption (DCP-ABE) scheme consists of the following five algorithms:

- Global Setup. This algorithm takes a security parameter λ as input and returns public parameters PP to the system.
- Authority Setup. This algorithm is run by each authority A_i to generate the relevant public key PK_i and secret key SK_i, where $i = 1, 2, \cdots, N$.
- KeyGen. Taking as input the public parameters PP, the secret keys SK_i, a user U's global identifier GID_U and a set of attributes $\tilde{U} \cap \tilde{A}_i$, this algorithm outputs a secret key SK_U^i. Here $\tilde{U}$ is used to indicate the set of attribute for user U, $\tilde{A}_i$ denotes the attributes monitored by the authority A_i.
- Encryption. It takes public parameters PP, a message $\mathcal{M}$, authority's public keys PK_i and an access structure W as input, returns the ciphertext CT.
- Decryption. Taking as input the global identifier GID, a collection of secret keys corresponding to user attributes and CT, then decryption algorithm outputs $\mathcal{M}$ when user attributes satisfy the access structure in ciphertext.

3.2 Security Model of DCP-ABE

Similar to [4,6,7], the selective access policy model is defined as follows.

- Instruction: The adversary $\mathcal{A}$ submits the set of challenge access structure W^* and a set of corrupted authorities $\mathfrak{U}$, where $|\mathfrak{U}| < N$.
- Global Setup: The challenger $\mathcal{B}$ runs the **Global Setup** algorithm and outputs the system parameters PP to $\mathcal{A}$.
- Authorities Setup: There are two different cases.
  (i) For the corrupted authority, $\mathcal{B}$ runs the **Authority Setup** algorithm to get the secret public key pair (PK_i, SK_i) and sends them to $\mathcal{A}$.
  (ii) For the uncorrupted authority, $\mathcal{B}$ runs the **Authority Setup** algorithm to get the secret public key pair (PK_i, SK_i) and sends PK_i to $\mathcal{A}$.
- Phase 1: $\mathcal{A}$ submits the user U^*'s attributes list L^* and global identifier GID_{U^*} to the challenger $\mathcal{B}$ for secret keys queries but $L^* \not\models W^*$. Then $\mathcal{B}$ runs the **KeyGen** algorithm and sends the corresponding SK_{U^*} to $\mathcal{A}$.
- Challenge: $\mathcal{A}$ submits two same-length messages $\mathcal{M}_0$, $\mathcal{M}_1$ and a challenge access structure W^* to $\mathcal{B}$. Then $\mathcal{B}$ flips an unbiased coin $\xi \in \{0, 1\}$ and runs the **Encryption** algorithm to encrypt $\mathcal{M}_\xi$ under access structure W^* and get the corresponding ciphertext CT^*. Finally, $\mathcal{B}$ sends CT^* to $\mathcal{A}$.
- Phase 2: Same as phase 1.
- Guess: Finally, $\mathcal{A}$ outputs the guess bit $\xi' \in \{0, 1\}$ for ξ and wins the game if $\xi' = \xi$.

Definition 1. *A DCP-ABE scheme is (t, q, ϵ) secure in the selective access policy model if all t-time adversary makes q secret key queries and succeeds in the above game with negligible advantage ϵ.*

3.3 Definition of Privacy-Preserving DCP-ABE

The definition of privacy-preserving DCP-ABE is similar as normal DCP-ABE except the KeyGen algorithm. In order to protect user privacy, the KeyGen algorithm is replaced by anonymous KeyGen algorithm in the DCP-ABE scheme. In the following, we will introduce the outline of our anonymous KeyGen algorithm.

– Anonymous KeyGen. The user U runs the commitment scheme proposed in Sect. 2.2, then he/she sends the *com* to authority A_i. From *com*, authority A_i can use the aforementioned zero-knowledge proof system in Sect. 2.3 to verify whether the user U has GID_U or not. If the proof is successful, A_i picks a random number $w_i^u \in_R \mathbb{Z}_p$ and computes partial secret keys for U. Again user U utilizes the aforementioned zero-knowledge proof system to verify whether these secret keys from A_i are correct or not. If the proof is successful and algorithm *Decommit* returns 1, the user U can compute his/her secret keys successfully and authority A_i gets empty. Otherwise, algorithm aborts and outputs $(\perp, \perp)$ for the authority and user.
To obtain the attribute secret keys, the anonymous KeyGen algorithm will employ the *k-out-of-n* oblivious transfer protocol introduced in Sect. 2.4. Before running the *k-out-of-n* oblivious transfer protocol, the user should convince authority A_i that he/she has the possession of attributes anonymous. In order to achieve this goal, we employ the anonymous credential system, which is proposed by Zhang and Feng [19]. In this anonymous credential system, the user can prove the possession of attributes without leaking any attribute information. If the user runs the anonymous credential system successfully, then the anonymous KeyGen algorithm will run the *k-out-of-n* oblivious transfer protocol to get the attribute secret key. From anonymous credential system and OT_n^k, the authority A_i can issue the correct attribute secret keys without knowing what attributes the user has. Firstly, the user prove the possession of attribute set $\tilde{U}$ anonymous by employing the anonymous credential system. If this anonymous credential system runs successfully, then the authority A_i takes $(pp, SK_i, \tilde{A}_i)$ as input and computes a set of attribute secret keys $\widetilde{SK}_{att}^{i}$. Finally, user U runs the OT_n^k and gets the attribute secret keys which ones are in $\tilde{U} \cap \tilde{A}_i$.

3.4 Security Model of Privacy-Preserving DCP-ABE

Following Han et al.'s scheme in [7], the security model of our privacy-preserving DCP-ABE is same as the model of DCP-ABE. Besides, the anonymous KeyGen algorithm should satisfy two extract properties: *leak-freeness* and *selective-failure blindness* [6,7]. Leak-freeness requires that a malicious user cannot learn anything which he/she cannot know by executing the anonymous KeyGen algorithm with an honest authority. Selective-failure blindness requires that a malicious authority cannot learn anything about user's identifier and his/her attributes. We will use the following two experiments to define the leak-freeness game.

– Real experiment: The distinguisher $\mathcal{D}$ runs the Global Setup algorithm and Authority Setup algorithm as many as he/she wants. The malicious user U with global identifier GID_U and a set of attributes $\tilde{U}$ executes the anonymous KeyGen algorithm with the honest authority A_i.
– Ideal experiment: The distinguisher $\mathcal{D}$ runs the Global Setup algorithm and Authority Setup algorithm as many as he/she wants. The malicious user U'

with global identifier $GID_{U'}$ and a set of attributes $\tilde{U}'$, and requires a trusted party to obtain the outputs of KeyGen algorithm.

Definition 2. *An anonymous KeyGen algorithm is leak-freeness if for any efficient adversary U, there exists a simulator U' such that no distinguisher $\mathcal{D}$ can distinguish whether U is playing in the real experiment or in the ideal experiment with non-negligible advantage.*

The selective-failure blindness game is defined as follows.

(i) The malicious authority A_i outputs its public key PK_i and two pairs of global identifiers and attribute sets $(GID_{U_0}, \tilde{U}_0)$ and $(GID_{U_1}, \tilde{U}_1)$.
(ii) Randomly choose a bit $b \in \{0, 1\}$.
(iii) A_i is given comments com_b and com_{1-b}. Then it black-box accesses oracles $\mathcal{U}(params, GID_{U_b}, \tilde{U}_b, PK_i, decom_b)$ and $\mathcal{U}(params, GID_{U_{1-b}}, \tilde{U}_{1-b}, PK_i, decom_{1-b})$.
(iv) The algorithm U outputs the secret keys $SK^i_{U_b}$ and $SK^i_{U_{1-b}}$, respectively.
(v) If $SK^i_{U_b} \neq \perp$ and $SK^i_{U_{1-b}} \neq \perp$, the A_i is given $(SK^i_{U_b}, SK^i_{U_{1-b}})$; if $SK^i_{U_b} \neq \perp$ and $SK^i_{U_{1-b}} = \perp$, the A_i is given $(\epsilon, \perp)$; if $SK^i_{U_b} = \perp$ and $SK^i_{U_{1-b}} \neq \perp$, the A_i is given $(\perp, \epsilon)$; if $SK^i_{U_b} = \perp$ and $SK^i_{U_{1-b}} = \perp$, the A_i is given (ϵ, ϵ).
(vi) Finally, A_i outputs its guess b' on b. A_i wins the game if $b' = b$.

Definition 3. *An anonymous KeyGen algorithm is selective-failure blindness if no probably polynomial time adversary A_i can win the above game with non-negligible advantage.*

Definition 4. *A privacy-preserving DCP-ABE scheme is secure if and only if it satisfies the following conditions:*

(i) The privacy-preserving DCP-ABE scheme is secure in the selective access policy model;
(ii) The anonymous KeyGen algorithm is both leak-freeness and selective-failure blindness.

4 Our Construction

4.1 Decentralized Ciphertext-Policy Attribute-Based Encryption

– Global Setup. To generate the global system parameters, this algorithm takes a security parameter λ as input. Then it returns a bilinear group $\Theta = (e, p, \mathbb{G}, \mathbb{G}_T)$ with prime order p. It chooses random generators $g, h, h_1 \in \mathbb{G}$ and the collision-resistant hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_p$ which takes the user U's global identifier GID_U as input. We denote the corresponding output by u. Suppose that there are N authorities $\{A_1, A_2, \cdots, A_N\}$ in the system and A_i monitors an attribute list $\tilde{A}_i = \{a_{i,1}, a_{i,2}, \cdots, a_{i,n_i}\}$, where $i = [1, N]$. Public parameters are $PP = \langle g, h, h_1, e, p, H, \mathbb{G}, \mathbb{G}_T \rangle$.

- Authorities Setup. Each authority A_i, where $i = [1, N]$, randomly chooses $\alpha_i, \beta_i \in_R \mathbb{Z}_p$ and $t_{i,j} \in_R \mathbb{Z}_p$ ($i = [1, n], j = [1, n_i]$). The authority computes $Y_i = e(g,g)^{\alpha_i}$, $E_i = g^{\beta_i}$ and $T_{i,j} = g^{t_{i,j}}$. Then A_i publishes the public keys $PK_i = \langle Y_i, E_i, \{T_{i,j}\}_{j=[1,n_i]} \rangle$ and keeps the master secret keys $SK_i = \langle \alpha_i, \beta_i, \{t_{i,j}\}_{j=[1,n_i]} \rangle$.
- KeyGen. To generate secret key for user U with GID_U and a set of attributes $\tilde{U}$, the authority A_i randomly picks $w^u_{i,j} \in_R \mathbb{Z}_p$ for each attribute $att_{i,j} \in \tilde{U} \cap \tilde{A}_i$ and sets $w^u_i = \Sigma_{att_{i,j} \in \tilde{U} \cap \tilde{A}_i} w^u_{i,j}$. Then A_i computes $D_{1,i} = g^{\alpha_i} h^{u\beta_i} h_1^{\beta_i w^u_i}$ and $D_{2,i,j} = h_1^{\frac{w^u_{i,j}\beta_i}{t_{i,j}}}$. Then the U's secret key is $SK^i_U = \langle D_{1,i}, D_{2,i,j} \rangle_{a_{i,j} \in \tilde{U} \cap \tilde{A}_i}$.
- Encryption. To encrypt a massage $\mathcal{M} \in \mathbb{G}_T$ under the access policy W, the encryptor randomly picks $s \in_R \mathbb{Z}_p$. Let $\mathcal{I}$ be a set which consists of the indexes of the authorities whose attributes are selected to $\mathcal{M}$. Then encryption algorithm computes $C_0 = \mathcal{M} \prod_{i \in \mathcal{I}} Y_i^s$, $C_1 = g^s$, $C_{2,i,j} = T^s_{i,j}$, $C_3 = \prod_{i \in \mathcal{I}} E_i^s$. The ciphertext is $CT = \langle C_0, C_1, C_{2,i,j}, C_3 \rangle_{a_{i,j} \in W}$.
- Decryption. To decrypt a ciphertext CT, the user who have a attribute set L can compute A, B, and E as follows if $L \models W$:

$$
\begin{aligned}
A &= \prod_{i \in \mathcal{I}} e(D_{1,i}, C_1) = \prod_{i \in \mathcal{I}} e(g^{\alpha_i} h^{u\beta_i} h_1^{\beta_i w^u_i}, g^s) \\
&= \prod_{i \in \mathcal{I}} e(g,g)^{\alpha_i s} \cdot \prod_{i \in \mathcal{I}} e(g,h)^{\beta_i u} \cdot \prod_{i \in \mathcal{I}} e(g, h_1)^{w^u_i \beta_i s}, \\
B &= \prod_{a_{i,j} \in W} e(D_{2,i,j}, C_{2,i,j}) = \prod_{a_{i,j} \in W} e(h_1^{\frac{w^u_{i,j}\beta_i}{t_{i,j}}}, g^{t_{i,j} s}) \\
&= \prod_{i \in \mathcal{I}} e(g, h_1)^{w^u_i \beta_i s}, \\
E &= e(h, C_3)^u = e(h, \prod_{i \in \mathcal{I}} g^{s\beta_i})^u = \prod_{i \in \mathcal{I}} e(g,h)^{us\beta_i}.
\end{aligned}
$$

  Therefore, the user can get the massage $\mathcal{M} = \frac{C_0 B E}{A}$.

4.2 Anonymous KeyGen Protocol

$\tilde{U}$. On the other hand authority A_i cannot learn anything about GID_U and what attribute the user has.

Algorithm 2 shows the anonymous KeyGen protocol, where the user U and authority A_i combine to compute partial decryption keys for U. Firstly, U randomly chooses $\rho \in_R$ then U and A_i interact with each other using two-party multiplication protocol 2MPC. Following the 2MPC protocol [14], it takes (u, ρ) from U and β_i from A_i as input and outputs $x = \rho u \beta_i \ mod \ p$ to A_i. Because ρ is randomly picked in $\mathbb{Z}p$ by U, A_i cannot learn anything about user's global identifier u from x. Next, U computes $P = g^{\frac{1}{\rho}}$, $Q = h_1^{\frac{1}{\rho}}$, $R = g^{\frac{1}{\rho^2}}$ and sends them to A_i. To proof that U knows (u, ρ) in zero-knowledge protocol, U computes $PoK\{(u, \rho) : \Psi = g^{u\rho}\}$ and sends it to A_i. After that, the authority A_i

Algorithm 2. Anonymous KeyGen Protocol

U randomly picks $\rho \in_R \mathbb{Z}p$
$U \overset{2MPC}{\longleftrightarrow} A_i : x = \rho u \beta_i$
U computes $PoK\{(u, \rho) : \Psi = g^{u\rho}\}$, $P = g^{\frac{1}{\rho}}$, $Q = h_1^{\frac{1}{\rho}}$, $R = h^{\frac{1}{\rho^2}}$ and sends them to A_i
A_i verifies $g^x \overset{?}{=} \Psi^{\beta_i}$
if $g^x = \Psi^{\beta_i}$ **then**
    A_i chooses a random number $w_i^u \in_R \mathbb{Z}p$. Then A_i computes $\widetilde{D_{1,i}} = P^{\alpha_i} Q^{w_i^u \beta_i} R^x$, $PoK\{(\alpha_i, w_i^u \beta_i, x) : \widetilde{D_{1,i}} = P^{\alpha_i} Q^{w_i^u \beta_i} R^x\}$ and sends them to U
else
    Abort
A_i proofs it knows $(\alpha_i, w_i^u \beta_i, x)$ in zero knowledge to U
if the proof is successful **then**
    U computes $D_{1,i} = (\widetilde{D_{1,i}})^{\rho}$
else
    Abort
$U \overset{OT_n^k}{\longleftrightarrow} A_i$: U gets $D_{2,i,j} = h_1^{\frac{w_{i,j}^u \beta_i}{t_{i,j}}}$

verifies $g^x \overset{?}{=} \Psi^{\beta_i}$, if it is correctly verified, A_i randomly chooses $w_{i,j}^u \in_R \mathbb{Z}_p$ for each attribute $att_{i,j} \in \tilde{U} \cap \tilde{A}_i$ and sets $w_i^u = \Sigma_{att_{i,j} \in \tilde{U} \cap \tilde{A}_i} w_{i,j}^u$ and computes $\widetilde{D_{1,i}} = P^{\alpha_i} Q^{w_i^u \beta_i} R^x$. Then authority A_i sends them to U. At the same time, A_i needs to prove he/she knows the three-tuple $(\alpha_i, w_i^u \beta_i, x)$ in zero-knowledge to U. The detailed steps as follows.

(i) A_i randomly picks $b_1, b_2, b_3 \in_R \mathbb{Z}p$, computes $\widetilde{D_{1,i}^*} = P^{b_1} Q^{b_2} R^{b_3}$ and sends $\widetilde{D_{1,i}}$, $\widetilde{D_{1,i}^*}$ to U.
(ii) U chooses $c \in_R \mathbb{Z}p$ and sends it to A_i.
(iii) A_i computes $b_1' = b_1 - c\alpha_i$, $b_2' = b_2 - c w_i^u \beta_i$, $b_3' = b_3 - cx$ and sends b_1', b_2', b_3' to U.
(iv) U verifies $\widetilde{D_{1,i}^*} \overset{?}{=} P^{b_1'} Q^{b_2'} R^{b_3'} (\widetilde{D_{1,i}})^c$.

If the proof is successful, U uses ρ to compute $D_{1,i} = (\widetilde{D_{1,i}})^{\rho}$. During the whole algorithm we can get user U's partial secret keys about global identifier without leaking any identity information. By employing *k-out-of-n* oblivious transfer protocol, attribute secret keys can be obtained anonymously.

In the proposed scheme, each authority A_i is in charge of a set of attribute $\tilde{A}_i = \{att_{i,1}, att_{i,2}, \cdots, att_{i,n_i}\}$. Firstly, A_i uses its secret keys $\beta_i, t_{i,j}, w_i^u$ and the public parameters h_1 to compute $\{h_1^{w_i^u \beta_i / t_{i,j}}\}_{a_{i,j} \in \tilde{A}_i}$. Recalling the OT_n^k protocol, we put $\{h_1^{w_i^u \beta_i / t_{i,j}}\}_{a_{i,j} \in \tilde{A}_i}$ as massages m_i and let U run this protocol for obtaining $m_{\sigma_1}, m_{\sigma_2}, \cdots, m_{\sigma_{|\tilde{U}|}}$. During this process, A_i cannot known which attribute secret keys are chosen by U, therefore the user U can gets his/her attribute secret keys $\{D_{2,i,j} = h_1^{w_i^u \beta_i / t_{i,j}}\}_{a_{i,j} \in \tilde{U} \cap \tilde{A}_i}$ anonymously. In conclusion, U obtains

his/her secret keys by interacting with A_i without leaking any information about GID_U and attributes to A_i.

5 Security of the Proposed Scheme

Theorem 1. *The proposed DCP-ABE is* (t, q, ϵ) *secure in the selective access policy model if all t-time adversary makes q secret key queries and succeeds in the following game with negligible advantage* ϵ.

Theorem 2. *The proposed anonymous KeyGen protocol is leak-free and selective-failure blind.*

Due to space limitations, detailed proofs of Theorems 1 and 2 will be given in the full version.

6 Performance Comparison

In this section, we will present the comparisons between previous different multi-authority CP-ABE schemes and ours with regard to security and efficiency.

Table 1. Security comparison among different multi-authority CP-ABE schemes.

Scheme	A_i cooperation	Attribute hidden	Tolerance	Access structure	Hardness
[7]	No	No	$N-1$	LSSS	q-PBDHE
[8]	Yes	Yes	$N-2$	Tree	DBDH
[13]	Yes	No	$N-2$	Threshold	DBDH
Ours	No	Yes	$N-1$	AND	DBDH

In Table 1, all of the schemes achieve the privacy protection for the global identifier, but only our scheme apparently achieves the privacy protection for the global identifier and attributes simultaneously. In addition, there is no cooperation among authorities in the initialization phase. Our construction is tolerant against maximum $(N-1)$ authorities colluding. And its security is reduced to a standard DBDH complexity assumption.

To simulate these schemes and get the computational costs, the Pairing-Based Cryptography (PBC) Library (version 0.5.14) is implemented. In the whole system, the PBC Library is implemented on a Windows machine with 2.67GHz Intel(R) Core(TM)2 Quad CPU and 4GB ROM. Without loss of generality, we assume that there are three authorities in these schemes. In Fig. 1(a), (c) and (d), the time complexity of all schemes increase linearly with the number of attributes in each authority. In Fig. 1(b), the key issuing time costs of [13] increases quadratically with the number of attributes because each pair of authorities must agree on a shared secret for a attribute. In Fig. 1, it is obvious to see that our scheme is efficient in computational costs comparing with among these schemes, especially in the key issuing phase.

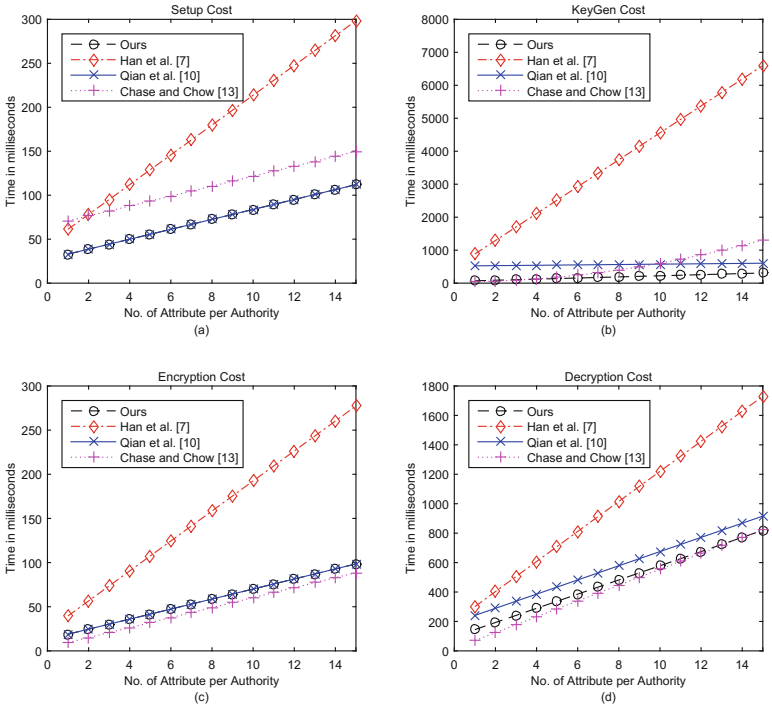

Fig. 1. Computational costs comparison among different multi-authority CP-ABE.

7 Conclusions

In this paper, we proposed a novel privacy-preserving DCP-ABE to answer the open problem in [12]. Due to our anonymous key generation protocol, the proposed scheme can protect user privacy by preventing authorities from learning any information about both GID and attributes. It requires neither a central authority nor cooperation among authorities for issuing secret keys. In addition, the proposed scheme achieves low computational costs and its security can be reduced to a standard DBDH assumption.

References

1. 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
2. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy, pp. 321–334. IEEE, Washington (2007)
3. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: 13th ACM Conference on Computer and Communications Security, pp. 89–98. ACM, New York (2006)

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. Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 568–588. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_31
6. Han, J., Susilo, W., Mu, Y., Yan, J.: Privacy-preserving decentralized key-policy attribute-based encryption. IEEE Trans. Parallel Distrib. Syst. **23**(11), 2150–2162 (2012)
7. Han, J., Susilo, W., Mu, Y., Zhou, J., Au, M.H.: PPDCP-ABE: privacy-preserving decentralized ciphertext-policy attribute-based encryption. In: Kutyłowski, M., Vaidya, J. (eds.) ESORICS 2014. LNCS, vol. 8713, pp. 73–90. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11212-1_5
8. Jung, T., Li, X.Y., Wan, Z., Wan, M.: Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption. IEEE Trans. Inf. Forensics Secur. **10**(1), 190–199 (2015)
9. Rahulamathavan, Y., Veluru, S., Han, J., Li, F., Rajarajan, M., Lu, R.: User collusion avoidance scheme for privacy-preserving decentralized key-policy attribute-based encryption. IEEE Trans. Comput. **65**(9), 2939–2946 (2016)
10. Qian, H., Li, J., Zhang, Y.: Privacy-preserving decentralized ciphertext-policy attribute-based encryption with fully hidden access structure. In: Qing, S., Zhou, J., Liu, D. (eds.) ICICS 2013. LNCS, vol. 8233, pp. 363–372. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-02726-5_26
11. Ge, A., Zhang, J., Zhang, R., Ma, C., Zhang, Z.: Security analysis of a privacy-preserving decentralized key-policy attribute-based encryption scheme. IEEE Trans. Parallel Distrib. Syst. **24**(11), 2319–2321 (2013)
12. Wang, M., Zhang, Z., Chen, C.: Security analysis of a privacy-preserving decentralized ciphertext-policy attribute-based encryption scheme. Concurr. Comput. Pract. Exp. **28**(4), 1237–1245 (2016)
13. Chase, M., Chow, S.S.M.: Improving privacy and security in multi-authority attribute-based encryption. In: 16th ACM Conference on Computer and Communications Security, pp. 121–130. ACM, New York (2009)
14. Lindell, Y., Pinkas, B.: Privacy preserving data mining. J. Cryptol. **15**(3), 177–206 (2002)
15. Li, J., Yao, W., Zhang, Y., Qian, H., Han, J.: Flexible and fine-grained attribute-based data storage in cloud computing. IEEE Trans. Serv. Comput. **10**(5), 785–796 (2017)
16. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_9
17. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups. In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 410–424. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052252
18. Chu, C.-K., Tzeng, W.-G.: Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 172–183. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30580-4_12
19. Zhang, Y., Feng, D.: Efficient attribute proofs in anonymous credential using attribute-based cryptography. In: Chim, T.W., Yuen, T.H. (eds.) ICICS 2012. LNCS, vol. 7618, pp. 408–415. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34129-8_39