IEEE Transactions on Information Forensics and Security,(Volume:10 , Issue: 3),March 2015.

Improving Privacy and Security in Decentralized Ciphertext-Policy Attribute-Based Encryption

Jinguang Han, *Member, IEEE,* Willy Susilo, *Senior Member, IEEE,* Yi Mu, *Senior Member, IEEE,* Jianying Zhou, and Man Ho Au, *Member, IEEE*

***Abstract*—In previous privacy-preserving multi-authority attribute-based encryption (PPMA-ABE) schemes, a user can acquire secret keys from multiple authorities with them knowing his/her attributes and furthermore, a central authority is required. Notably, a user's identity information can be extracted from his/her some sensitive attributes. Hence, existing PPMA-ABE schemes cannot fully protect users' privacy as multiple authorities can collaborate to identify a user by collecting and analyzing his attributes. Moreover, ciphertext-policy ABE (CP-ABE) is a more efficient public-key encryption where the encryptor can select flexible access structures to encrypt messages. Therefore, a challenging and important work is to construct a PPMA-ABE scheme where there is no necessity of having the central authority and furthermore, both the identifiers and the attributes can be protected to be known by the authorities. In this paper, a privacy-preserving decentralized CP-ABE (PPDCP-ABE) is proposed to reduce the trust on the central authority and protect users' privacy. In our PPDCP-ABE scheme, each authority can work independently without any collaboration to initial the system and issue secret keys to users. Furthermore, a user can obtain secret keys from multiple authorities without them knowing anything about his global identifier (GID) and attributes.**

***Index Terms*—CP-ABE, decentralization, privacy**

I. INTRODUCTION

IN network society, attributes are used to distinguish different users. For instance, European electronic identity cards often comprise the attributes: nationality, sex, civil status, hair and eye color, and applicable minority status. These attributes can be either binary or discrete numbers from a pre-defined finite sets [2]. In particular, these attributes are required to selectively disclose as they are privacy-sensitive; otherwise, a user can be identified and impersonated if some of his/her sensitive attributes are collected.

In practice, we often want to share data with some expressive attributes and do not know who the recipient will be. To resolve this problem, a new public-key encryption system called attribute-based encryption (ABE) was introduced in the seminal work of Sahai and Waters [3]. In an ABE scheme, there is a central authority who monitors a set of universal attributes and issues secret keys to users accordingly. As a result, a user can decrypt a ciphertext if and only if there is a match between the attributes which are listed in the ciphertext and the attributes which he holds. ABE schemes have been the primary focus in the research community nowadays as it allows flexible access control and can protect the confidentiality of sensitive data [4], [5], [6], [7], [8], [9].

In an ABE scheme [3], a central authority is required. To reduce the trust on the central authority, Chase [10] proposed a multi-authority ABE (MA-ABE) scheme. In this scheme, multiple authorities can co-exist and must cooperate with the central authority to initialize the system. Then, Lewko and Waters [11] proposed a decentralized CP-ABE (DCP-ABE) where a central authority is not required and multiple authorities can work independently without any cooperation.

Since the authorities can impersonate a user if they can know his attributes, privacy issues in MA-ABE are the primary concern of users. Considering this issue, some schemes have been proposed, but they cannot provide a complete solution. In all the previous privacy-preserving MA-ABE (PPMA-ABE) schemes [12], [13], [14], only the privacy of the global identifier (GID) has been considered. Currently, no scheme addressing the privacy of the attributes in MA-ABE has been proposed. However, it is extremely important as a user can be identified by some sensitive attributes. To clarify this, we give the following example. Suppose that the Head of the Department of Computer Science is Bob. Given two sets of attributes S_1={Position="Head", Department="CS", Sex="Male"} and S_2= {Position="PhD Student", Department="CS", Sex="Male"}, we can guess that S_1 is Bob's attributes even if we do not know his GID. This clearly shows that it is necessary to control the release of sensitive attributes.

Part of this paper appeared in The 19th European Symposium on Research in Computer Security (ESORICS 2014) [1]. In this revised version, the following contents have been added. First, we explain the challenge to construct a PPDCP-ABE, and then introduce our corresponding techniques in Section I-B. Second, the formal proofs of the proposed schemes are provided in Section IV-B and Section IV-F, respectively. Third, we give an instance to explain how the proposed privacy-preserving key extract protocol works in Section IV-E. Finally, the computation cost and communication cost of the proposed schemes are analyzed in Section IV-C and Section IV-G, respectively.

J. Han is with Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing, Jiangsu 210003, China. E-mail: jghan22@gmail.com

W. Susilo and Y. Mu are with School of Computer Science and Software Engineering, University of Wollongong, NSW2522, Australia. E-mail: wsusilo@uow.edu.au; ymu@uow.edu.au. The corresponding author is W. Susilo.

J. Zhou is with Infocomm Security Department, Institute for Infocomm Research, 1 Fusionopolis Way, Singapore 138632, Singapore. E-mail: jyzhou@i2r.a-star.edu.sg

M. H. Au is with Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. E-mail: csallen@comp.polyu.edu.hk

A. Our Contributions

In this paper, we propose a privacy-preserving DCP-ABE (PPDCP-ABE) scheme where the central authority is not required and each authority can work independently without any cooperation. As a notable feature, each authority can dynamically join or leave the system, namely other authorities do not need to change their secret keys and reinitialize the system when an authority joins or leaves the system. Each authority monitors a set of attributes and issues secret keys to users accordingly. To resist the collusion attacks, a user's secret keys are tied to his GID. Especially, a user can obtain secret keys for his attributes from multiple authorities without them knowing any information about his GID and attributes. Therefore, the proposed PPDCP-ABE scheme can provide stronger privacy protection compared to the previous PPMA-ABE schemes where only the GID is protected.

When encrypting a message, the encryptor can select an access structure for each authority and encrypt the message under the selected access structures so that a user can decrypt the ciphertext if his attributes satisfy all the access structures. Comparatively, our scheme is constructed in the standard model, while the existing DCP-ABE scheme [11] was designed in the random oracle model. To the best of our knowledge, it is the *first* PPDCP-ABE scheme where the privacy of both the identifiers and attributes are considered.

B. Challenges and Techniques

Challenge. When constructing a PPDCP-ABE scheme, the following technical hurdles must be overcome.

First, the collusion attacks must be resisted. Since the DCP-ABE scheme [11] was constructed in the radome oracle model, the collusion attacks can be easily resisted by tieing the user's secret keys to his GID. However, it is challenging to resist the collusion attacks in the DCP-ABE scheme which is designed in the standard model;

Second, the user must convince each authority that the attributes for which he is obtaining secret keys are monitored by the authority as the authority cannot know his attributes;

Third, the authority can interact with the user to generate correct secret keys for him even if he dose not know the user's identifer and attributes;

Finally, the secret keys derived from multiple authorities can be used together to decrypt a ciphertext.

Techniques. To overcome the hurdles mentioned above, the following techniques are exploited.

In [11], to resist the collusion attacks, each authority A_i ties a user's secret keys to his GID by computing $H(GID)^{y_i}$ where y_i is A_i's secret key and $H(\cdot)$ is a hash function. In the standard model, when creating secret keys for a user, each authority selects a random number t and computes $\mathfrak{g}^t\mathfrak{g}^{\frac{\beta+\mu}{t}}$ where $\mathfrak{g}$ is the generator of a group $\mathbb{G}$, β is the partial master secret key of the authority and μ is the user's identifier. Therefore, the secret keys generated for different users cannot be combined.

For the second problem, we exploit the set-membership proof technique. For each attribute, the authority specifies an unforgeable authentication tag such that a user can prove in zero knowledge that the attribute for which he is possessing a secret key is monitored by the authority.

To resolve the third problem, we use the idea in the CP-ABE scheme [9] and 2-party secure computing technique. In the traditional ABE schemes, for each attribute, the authority selects a secret key r and publishes the corresponding public key g^r. Then, the authority must use r [4], [11] or $\frac{1}{r}$ [3], [6], [10], [12], [15], [13] to generate a secret key for the attribute. However, this technique is not suitable to our scenario as the authority cannot know the user's attributes. We use the technique introduced in [9] where, for each attribute, the authority selects a random element from the group as the public key. To generate secret keys for a set of attributes, the authority selects a random number and computes the secret keys by randomizing the corresponding public keys. Hence, by using this technique, the user is allowed to first commit the public keys, then execute 2-party secure computing protocols with the authority to obtain the corresponding secret keys for his attributes.

Finally, we resolve the fourth problem by splitting the secret number used to encrypt a message into multiple parts. Each part is shared by an access structure. If all the access structures can be satisfied by the user's attributes, he can reconstruct all parts of the secret number and decrypt the ciphertext.

C. Organization

In Section II, we introduce the related work. The preliminaries which are used throughout this paper is introduced in Section III. In Section IV, we first propose a DCP-ABE scheme, and prove its security. Subsequently, we propose a privacy-preserving key extract algorithm and prove its security. Finally, we conclude this paper in Section V.

II. Related work

The related work is introduced in this section.

A. Attribute-based Encryption

Sahai and Waters [3] introduced the first attribute-based encryption (ABE) where both the ciphertext and the secret key are labeled with a set of attributes. A user can decrypt a ciphertext if and only if there is a match between the attributes listed in the ciphertext and the attributes held by him. ABE schemes can be classified into two types: key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE).

KP-ABE. In a KP-ABE scheme, the ciphertext is associated with a set of attributes, while an access structure is embedded in the secret keys [3], [10], [12], [6], [7], [13].

CP-ABE. In a CP-ABE scheme, an access structure is embedded in the ciphertext, while the secret keys are associated with a set of attributes [4], [5], [16].

B. Multi-Authority Attribute-based Encryption

In the seminal work [3], Sahai and Waters left an open problem, namely how to construct an ABE scheme where the secret keys can be extracted from multiple authorities so that users can reduce the trust on the central authority. Chase [10] answered this question affirmatively by proposing an MA-ABE scheme. As mentioned in [10], the technical hurdle in constructing an MA-ABE scheme is to resist the collusion attacks. To overcome this hurdle, all secret keys of a user are tied to his GID. In [10], multiple authorities must interact to initialize the system, and a central authority is required.

Lin *et al.* [17] proposed an MA-ABE scheme where the cental authority is not required. This scheme was derived from the distributed key generation (DKG) protocol [18] and the joint zero secret sharing (JZSS) protocol [19]. To initialize the system, the multiple authorities must collaboratively execute the DKG protocol and the JZSS protocol twice and k times, respectively, where k is the degree of the polynomial selected by each authority. Each authority must keep $k+2$ secret keys. Furthermore, this scheme is k-resilient, namely the scheme is secure if and only if the number of the compromised users is no more than k, and k must be fixed in the setup stage.

Mϋller *et al.* [20] proposed a distributed CP-ABE scheme. This scheme was proven to be secure in the generic group [4], instead of reducing to a complexity assumption. In this scheme, a central authority is required to generate the global key and issue secret keys to users.

A fully secure multi-authority CP-ABE (MACP-ABE) scheme in the standard model was proposed by Liu *et al.* [21]. This scheme was based on the previous CP-ABE scheme [8]. In this scheme, there are multiple central authorities and attribute authorities. The central authorities distribute identity-related keys to users, while the attribute authorities distribute attribute-related keys to users. Prior to possessing attribute keys from the attribute authorities, the user must obtain secret keys from the multiple central authorities. This scheme was constructed in the bilinear group with composite order $(N = p_1p_2p_3)$.

Lekwo and Waters [11] proposed a new MA-ABE scheme called decentralizing CP-ABE (DCP-ABE) scheme. This scheme improved the previous MA-ABE schemes that require collaborations among multiple authorities to initial the system. In this scheme, no cooperation between the multiple authorities is required in the setup stage and the key generation stage, and a central authority is not required. Notably, an authority in this scheme can join or leave the system dynamically without the need to reinitialize the system. The scheme was constructed in the bilinear group with composite order $(N = p_1p_2p_3)$, and achieved full (adaptive) security in the random oracle model. Furthermore, they also proposed two methods to create a prime order group variant of their scheme. Nevertheless, the authorities can collect a user's attributes by tracing his GID.

Chase and Chow first proposed [12] a privacy-preserving MA-ABE (PPMA-ABE) scheme which improved the previous scheme [10] and removed the need of a central authority. In previous MA-ABE schemes [10], [17], to obtain the corresponding secret keys, a user must submit his GID to each authority. Hence, multiple authorities can collaborate to collect the user's attributes by his GID. In [12], Chase and Chow provided an anonymous key issuing protocol for the GID by usinge the 2-party secure computing technique. As a result, a group of authorities cannot collaborate to collect the users attributes by tracing his GID. Nevertheless, the multiple authorities must cooperate to initial the system. Meanwhile, each pair of authorities must execute the 2-party key exchange protocol to share the seeds of the selected pseudo random functions (PRFs) [22]. This scheme is $N-2$ tolerant, namely the scheme is secure if and only if the number of the compromised authorities is no more than $N-2$, where N is the number of the authorities in the system. The authorities cannot know any information about the user's GID, but they can know the user's attributes. Chase and Chow [12] also left an open challenging research problem on how to construct a PPMA-ABE scheme without the need of cooperations among authorities.

Li [15] proposed a MACP-ABE scheme with accountability. In this scheme, the anonymous key issuing protocol [12] was employed. Specifically, a user can be identified when he shared his secret keys with others. Likewise, the multiple authorities must cooperate to initialize the system.

Recently, a privacy-preserving decentralized KP-ABE (PPDKP-ABE) scheme was proposed by Han *et al.* [13]. In this scheme, multiple authorities can work independently without any collaboration. Especially, a user can obtain secret keys from multiple authorities without releasing anything about his GID to them, and the central authority is not required. Qian *et al.* [14] proposed a privacy-preserving decentralized CP-ABE (PPDCP-ABE) scheme where simple access structures can be implemented. Nevertheless, similar to that in [12], the authorities in these schemes can also collect the user's attributes.

C. Anonymous Credential

In an anonymous credential system [23], a user can obtain a credential from an issuer, which includes the user's pseudonym and attributes. By using it, the user can convince a third party that he obtains a credential containing the given pseudonym and attributes without releasing any other information. In a multiple-show credential system [24], a credential can be demonstrated an arbitrary number of times, and cannot be linked to each other.

Therefore, when constructing our PPDCP-ABE, we assume that each user has obtained an anonymous credential including his GID and attributes. Then, he can convince the multiple authorities that he has a GID and holds the corresponding attributes by using the anonymous credential technique.

III. Preliminaries

In this section, the preliminaries used throughout this paper is introduced.

A function $\epsilon : \mathbb{Z} \rightarrow R$ is negligible if for any $z \in \mathbb{Z}$ there exists a k such that $\epsilon(x) < \frac{1}{x^z}$ when $x > k$. By $\mathcal{KG}(1^\kappa) \rightarrow (SK, PK)$, we denote a secret-public key pair generator which takes as input a security parameter 1^κ and outputs a secret-public key pair (SK, PK). Unless otherwise specified, by $\alpha \stackrel{\$}{\leftarrow} A$, we denote that α is selected from A randomly. Especially, $\alpha \stackrel{\$}{\leftarrow} A$ stands for that α is selected from A uniformly at random if A is a finite set. $|A|$ stands for the cardinality of a finite set A. By $A(x) \rightarrow y$, we denote that y is computed by running the algorithm A with input x. We use $\mathbb{Z}_p$ to denote a finite field with prime order p. Finally, $R \stackrel{r}{\rightarrow} S$ and $R \stackrel{s}{\leftarrow} S$ are used to denote that the party R sends r to the party S and the party S sends s to the party R, respectively. $U_1 \bigcap U_2$ and $U_1 \bigcup U_2$ stand for the intersection and union of the sets U_1 and U_2, respectively.

A. Complexity Assumption

Let $\mathbb{G}$ and $\mathbb{G}_\tau$ be two cyclic groups with prime order p, and g be a generator of $\mathbb{G}$. A map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\tau$ is a bilinear map if the following properties can be satisfied:

1) Bilinearity. For all $x, y \in \mathbb{Z}_p$ and $u, v \in \mathbb{G}$, $e(u^x, v^y) = e(u^y, v^x) = e(u, v)^{xy}$.
2) Nondegeneracy. $e(g, g) \neq 1_\tau$ where 1_τ is the identity of the group $\mathbb{G}_\tau$.
3) Computability. For all $u, v \in \mathbb{G}$, there exists an efficient algorithm to compute $e(u, v)$.

$\mathcal{GG}(1^\kappa) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$ stands for a bilinear group generator which takes as input a security parameter 1^κ and outputs a bilinear group $(e, p, \mathbb{G}, \mathbb{G}_\tau)$ with prime order p and a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\tau$. By $TE_{\mathbb{G}}, TE_{\mathbb{G}_\tau}, TP$, we denote the running time of computing an exponential on $\mathbb{G}$, the running time of computing an exponential on $\mathbb{G}_\tau$ and the running time of computing a pairing, respectively. By $E_{\mathbb{G}}$ and $E_{\mathbb{G}_\tau}$, we denote the length of one element in $\mathbb{G}$ and $\mathbb{G}_\tau$, respectively.

Definition 1. (q-Strong Diffie-Hellman (q-SDH) Assumption [25]) *Suppose that $x \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, $\mathcal{GG}(1^\kappa) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$ and g is a generator of $\mathbb{G}$. Given a $(q+1)$-tuple $\overrightarrow{y} = (g, g^x, g^{x^2}, \cdots, g^{x^q})$, we say that the q-SDH assumption holds on the bilinear group $(e, p, \mathbb{G}, \mathbb{G}_\tau)$ if no probabilistic polynomial-time adversary $\mathcal{A}$ can output $(c, g^{\frac{1}{x+c}})$ with the advantage*

$$Adv_{\mathcal{A}} = \Pr[\mathcal{A}(\overrightarrow{y}) \rightarrow (c, g^{\frac{1}{x+c}})] \geq \epsilon(k)$$

where $c \in \mathbb{Z}_p^*$ and the probability is token over the random choices $x \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ and the random bits consumed by $\mathcal{A}$.

Definition 2. (Decisional q-Parallel Bilinear Diffie-Hellman Exponent (q-PBDHE) Assumption [9]) *Suppose that $a, s, b_1, \cdots, b_q \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, $\mathcal{GG}(1^\kappa) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$ and g is a generator of $\mathbb{G}$. Given a tuple $\overrightarrow{y} =$*

$$g, g^s, g^a, \cdots, g^{(a^q)}, g^{(a^{q+2})}, \cdots, g^{(a^{2q})}$$
$$\forall_{1 \leq j \leq q} \quad g^{s \cdot b_j}, g^{\frac{a}{b_j}}, \cdots, g^{(\frac{a^q}{b_j})}, g^{(\frac{a^{q+2}}{b_j})}, \cdots, g^{(\frac{a^{2q}}{b_j})}$$
$$\forall_{1 \leq j,k \leq q, k \neq j} \quad g^{\frac{a \cdot s \cdot b_k}{b_j}}, \cdots, g^{(\frac{a^q \cdot s \cdot b_k}{b_j})},$$

we say that the decisional q-PBDHE assumption hold on the bilinear group $(e, p, \mathbb{G}, \mathbb{G}_\tau)$ if no probabilistic polynomial-time adversary $\mathcal{A}$ can distinguish $(\overrightarrow{y}, e(g, g)^{a^{q+1}s})$ from $(\overrightarrow{y}, R)$ with the advantage

$$Adv_{\mathcal{A}} = \left| \Pr[\mathcal{A}(\overrightarrow{y}, e(g, g)^{a^{q+1}s}) = 1] - \Pr[\mathcal{A}(\overrightarrow{y}, R) = 1] \right| \geq \epsilon(k),$$

where $R \stackrel{\$}{\leftarrow} \mathbb{G}_\tau$ and the probability is token over the random choices of $a, s, b_1, \cdots, b_q \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ and the bits consumed by $\mathcal{A}$.

B. Building Blocks

To construct a PPDCP-ABE scheme, the following building blocks are adopted.

Definition 3. (Access Structure [26]) *Let $\mathcal{P} = (P_1, P_2, \cdots, P_n)$ be n parties. A collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots, P_n\}}$ is monotonic if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. An access structure (respectively monotonic access structure) is a collection (respectively monotonic collection) $\mathbb{A}$ of the non-empty subset of $(P_1, P_2, \cdots, P_n)$, i.e., $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots, P_n\}} \setminus \{\phi\}$. A set P is called an authorized set if $P \in \mathbb{A}$; otherwise P is an unauthorized set.*

Definition 4. (Linear Secret Sharing Schemes [26]) *A secret sharing scheme $\prod$ over a set of parties $\mathcal{P}$ is called linear (over $\mathbb{Z}_p$) if the following properties can be satisfied:*

1) *The shares for each party form a vector over $\mathbb{Z}_p$.*
2) *For $\prod$, there exists a matrix M with ℓ rows and n columns called the share-generating matrix. For $i = 1, 2, \cdots, \ell$, the ith row is labeled with a party $\rho(i)$ where $\rho : \{1, 2, \cdots, \ell\} \rightarrow \mathbb{Z}_p$. To share a secret $s \in \mathbb{Z}_p$, a vector $\overrightarrow{v} = (s, v_2, \cdots, v_n)$ is selected, where $v_2, \cdots, v_n$ are randomly selected from $\mathbb{Z}_p$. $M\overrightarrow{v}$ is the vector of the ℓ shares according to $\prod$. The share $M_i\overrightarrow{v}$ belongs to the party $\rho(i)$, where M_i is the ith row of M.*

Linear reconstruction property. Let S be an authorized set and $\mathcal{I} = \{i | \rho(i) \in S\}$. Then, there exists a set of constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in \mathcal{I}}$ such that, for any valid shares λ_i according to $\prod$, $\sum_{i \in \mathcal{I}} \omega_i \lambda_i = s$. $\{\omega_i\}_{i \in \mathcal{I}}$ can be computed in polynomial time with the size of share-generating matrix M.

Commitment Schemes. A commitment scheme consists of the following three algorithms.

Setup$(1^\kappa) \rightarrow params$. Taking as input a security parameter 1^κ, this algorithm outputs the public parameters $params$.

Commit$(params, m) \rightarrow (com, decom)$. Taking as input the public parameters $params$ and a message m, this algorithm outputs a commitment com and a decommitment $decom$. $decom$ can be used to decommit com to m.

Decommit$(params, m, com, decom) \rightarrow \{0, 1\}$. Taking as input the public parameters $params$, the message m, the commitment com and the decommitment $decom$, this algorithm outputs 1 if $decom$ can decommit com to m; otherwise, it outputs 0.

A commitment scheme must exhibit two properties: *hiding* and *binding*. The hiding property requires that the message

IEEE Transactions on Information Forensics and Security,(Volume:10 , Issue: 3),March 2015.

m keeps unreleased until the user releases it later, while the binding property requires that only the value $decom$ can be used to decommit the commitment com to m.

In this paper, we use the Pedersen commitment scheme [27] which is a perfectly hiding commitment scheme and is based on the discrete logarithm assumption. This scheme can be described as follows. Suppose that $\mathbb{G}$ is a cyclic group with prime order p, and $g_0, g_1, \cdots, g_k$ are generators of $\mathbb{G}$. To commit a tuple of messages $(m_1, m_2, \cdots, m_k)$, the user selects $r \xleftarrow{\$} \mathbb{Z}_p$, and computes $R = g_0^r g_1^{m_1} g_2^{m_2} \cdots g_k^{m_k}$. Then, the user can use r to decommit the commitment R.

Proof of Knowledge. We use the notion introduced by Camenisch and Stadler [28] to prove statements about discrete logarithm. By PoK $\left\{(\alpha, \beta, \gamma) : y = g^{\alpha} h^{\beta} \wedge \tilde{y} = \tilde{g}^{\alpha} \tilde{h}^{\gamma}\right\}$, we denote a zero knowledge proof of knowledge of integers α, β and γ such that $y = g^{\alpha} h^{\beta}$ and $\tilde{y} = \tilde{g}^{\alpha} \tilde{h}^{\gamma}$ hold on the group $\mathbb{G} = \langle g \rangle = \langle h \rangle$ and $\tilde{\mathbb{G}} = \langle g \rangle = \langle h \rangle$, respectively. Conventionally, the values in the parenthesis denote the knowledge that is being proven, while the rest of the values are known by the verifier. Notably, there exists an efficient extractor that can be used to rewind the knowledge from the successful prover.

Set-Membership Proof. Camenisch *et al.* [29] proposed a set membership proof scheme. This scheme is as follows. Let $\mathcal{GG}(1^{\kappa}) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_{\tau})$, and g, h be generators of $\mathbb{G}$.

1) Suppose that $\Phi \subseteq \mathbb{Z}_p$ is a finite set, for $i \in \Phi$, the verifier picks up $x \xleftarrow{\$} \mathbb{Z}_p$, and computes $Y = g^x$ and $T_i = g^{\frac{1}{x+i}}$. Then, it sends $\{Y, (T_i)_{i \in \Phi}\}$ to the prover.
2) To prove $\sigma \in \Phi$, the prover chooses $v, s, t, r, k \xleftarrow{\$} \mathbb{Z}_p$, and computes $C = g^{\sigma} h^r$, $D = g^s h^k$, $V = g^{\frac{v}{x+\sigma}}$ and $A = e(V, g)^{-s} \cdot e(g, g)^t$. Then, it sends (C, D, V, A) to the verifier.
3) The verifier selects $c \xleftarrow{\$} \mathbb{Z}_p$, and sends it to the prover.
4) The prover computes $z_{\sigma} = s - c\sigma$, $z_r = k - cr$ and $z_v = t - cv$, and sends (z_{σ}, z_k, z_t) to the verifier.
5) The verifier verifies $D \stackrel{?}{=} C^c g^{z_{\sigma}} h^{z_r}$ and $A \stackrel{?}{=} e(Y, v)^c \cdot e(V, g)^{-z_{\sigma}} \cdot e(g, g)^{z_r}$.

Theorem 1. *This protocol is a zero-knowledge argument of set-membership proof for a set Φ if the $|\Phi|$-SDH assumption holds on the bilinear group $(e, p, \mathbb{G}, \mathbb{G}_{\tau})$ [29].*

C. DCP-ABE: Decentralized Ciphertext-Policy Attribute-Based Encryption

A DCP-ABE scheme comprises the following algorithms.

Global Setup$(1^{\kappa}) \rightarrow params$. Taking as input a security parameter 1^{κ}, the global setup algorithm outputs the public parameter $params$. Suppose that there are N authorities $\{\breve{A}_1, \breve{A}_2, \cdots, \breve{A}_N\}$, and each authority $\breve{A}_i$ monitors a set of attributes $\tilde{A}_i$. Each user U has an unique global identifier GID_U and holds a set of attributes $\tilde{U}$.

Authority Setup$(1^{\kappa}) \rightarrow (SK_i, PK_i)$. Taking as input the security parameter 1^{κ}, the authority setup algorithm outputs a secret-public key pair (SK_i, PK_i) for each authority $\breve{A}_i$, where $\mathcal{KG}(1^{\kappa}) \rightarrow (SK_i, PK_i)$.

Encrypt$(params, \mathcal{M}, (M_i, \rho_i, PK_i)_{i \in \mathcal{I}}) \rightarrow CT$. Taking as input the public parameter $params$, a message $\mathcal{M}$, a set of access structures $(M_i, \rho_i)_{i \in \mathcal{I}}$ and a set of public keys $(PK_i)_{i \in \mathcal{I}}$, the encryption algorithm outputs the ciphertext CT.

KeyGen$(params, SK_i, GID_U, \tilde{U} \bigcap \tilde{A}_i) \rightarrow SK_U^i$. Taking as input the public parameter $params$, the secret key SK_i, a user's global identifier GID_U and a set of attributes $\tilde{U} \bigcap \tilde{A}_i$, the key generation algorithm outputs a secret key SK_U^i for U.

Decrypt$(params, GID, (SK_U^i)_{i \in \mathcal{I}}, CT) \rightarrow \mathcal{M}$. Taking as input the public parameter $params$, the user's globe identifier GID_U, the secret keys $(SK_U^i)_{i \in \mathcal{I}}$ and the ciphertext CT, the decryption algorithm outputs the message $\mathcal{M}$.

Definition 5. *A decentralized ciphertext-policy attribute-based encryption (DCP-ABE) is correct if*

$$\Pr\left[\begin{array}{l|l} & \text{Global Setup}(1^{\kappa}) \rightarrow \\ & params; \\ & \text{Authority Setup}(1^{\kappa}) \rightarrow \\ \text{Decrypt}(params, & (SK_i, Pk_i); \\ GID, (SK_U^i)_{i \in \mathcal{I}}, & \text{Encrypt}(params, \mathcal{M}, (M_i, \\ CT) \rightarrow \mathcal{M} & \rho_i, PK_i)_{i \in \mathcal{I}}) \rightarrow CT; \\ & \text{KeyGen}(params, SK_i, \\ & GID_U, \tilde{U} \bigcap \tilde{A}_i) \rightarrow SK_U^i \end{array}\right] = 1$$

where the probability is token over the random bits consumed by all the algorithms in the scheme.

D. Security Model of Decentralized Ciphertext-Policy Attribute-Based Encryption

This model is named as selective-access structure model, and is similar to that introduced in [10], [12], [13], [11], [9].

Initialization. The adversary $\mathcal{A}$ submits a list of corrupted authorities $\mathfrak{A} = \{\breve{A}_i\}_{i \in \mathcal{I}}$ and a set of access structures $\mathbb{A} = \{M_i^*, \rho_i^*\}_{i \in \mathcal{I}^*}$, where $\mathcal{I} \subseteq \{1, 2, \cdots, N\}$ and $\mathcal{I}^* \subseteq \{1, 2, \cdots, N\}$. There should be at least an access structure $(M^*, \rho^*) \in \mathbb{A}$ which cannot be satisfied by the attributes selected by $\mathcal{A}$ to query secrete keys and the attributes monitored by the authorities in $\mathfrak{A}$.

Global Setup. The challenger runs the Global Setup algorithm to generate the public parameters $params$, and sends them to $\mathcal{A}$.

Authority Setup. There are two cases.

1) For the authority $\breve{A}_i \subseteq \mathfrak{A}$, the challenger runs the Authority Setup algorithm to generate the secret-public key pair (SK_i, PK_i), and sends them to $\mathcal{A}$.
2) For the authority $\breve{A}_i \not\subseteq \mathfrak{A}$, the challenger runs the Authority Setup algorithm to generate the secret-public key pair (SK_i, PK_i), and sends the public key PK_i to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ can query secret key for a user U with an identifier GID_U and a set of attributes $\tilde{U}$. The challenger runs the KeyGen algorithm to generate a secret key SK_U, and sends it to $\mathcal{A}$. This query can be made adaptively and repeatedly.

Challenge. $\mathcal{A}$ submits two messages $\mathcal{M}_0$ and $\mathcal{M}_1$ with the same length. The challenger flips an unbiased coin with $\{0,1\}$, and obtains a bit $b \in \{0,1\}$. Then, the challenger runs Encrypt$(parmas, \mathcal{M}_b, (M_i^*, \rho^*, PK_i)_{i\in\mathcal{I}^*})$ to generate the challenged ciphertext CT^*, and then sends CT^* to $\mathcal{A}$.

Phase 2. Phase 1 is repeated.

Guess. Finally, $\mathcal{A}$ outputs his guess b' on b. $\mathcal{A}$ wins the game if $b' = b$.

Definition 6. (Selective-Access Structure Secure DCP-ABE (IND-sAS-CPA)) *A decentralized ciphertext-policy attribute-based encryption (DCP-ABE) scheme is* $(T, q, \epsilon(\kappa))$ *secure in the selective-access structure model if no probably polynomial-time adversary* $\mathcal{A}$ *making* q *secret key queries can win the above game with the advantage*

$$Adv_{\mathcal{A}}^{DCP-ABE} = \left|\Pr[b' = b] - \frac{1}{2}\right| > \epsilon(\kappa)$$

where the probability is token over all the bits consumed by the challenger and the adversary.

E. PPDCP-ABE: Privacy-Preserving Decentralized Ciphertext-Policy Attribute-Based Encryption

A PPDCP-ABE has the same algorithms Global Setup, Authority Setup, Encrypt and Decrypt with the DCP-ABE scheme. The main difference lies in that we replace the KeyGen algorithm with a privacy-preserving key generation algorithm PPKeyGen. Considering privacy issues, the authorities should not know both the user's identifier and attributes in a PPDCP-ABE scheme. This is motivated by the blind IBE schemes [30], [31]. The PPKeyGen algorithm is formalized as follows.

PPKeyGen$(U(params, GID_U, \tilde{U}, PK_i, decom_i, (decom_{i,j})_{a_{i,j}\in\tilde{U}\bigcap\tilde{A}_i}) \leftrightarrow \breve{A}_i(params, SK_i, PK_i, com_i, (com_{i,j})_{a_{i,j}\in\bigcap\tilde{A}_i})) \rightarrow (SK_U^i, \text{empty})$. This is an interactive algorithm executed between a user U and an authority $\breve{A}_i$. U runs the commitment algorithm Commit$(params, GID_U) \rightarrow (com_i, decom_i)$ and Commit$(params, a_{i,j}) \rightarrow (com_{i,j}, decom_{i,j})$ for the attribute $a_{i,j} \in \tilde{U}\bigcap\tilde{A}_i$, and sends $(com_i, (com_{i,j})_{a_{i,j}\in\bigcap\tilde{A}_i})$ to the authority $\breve{A}_i$. Then, U and $\breve{A}_i$ take as input $(params, GID_U, \tilde{U}, PK_i, decom_i, (decom_{i,j})_{a_{i,j}\in\tilde{U}\bigcap\tilde{A}_i})$ and $(params, SK_i, PK_i, com_i, (com_{i,j})_{a_{i,j}\in\bigcap\tilde{A}_i})$, respectively. If Decommit$(params, GID_U, com_i, dcom_i) = 1$ and Decommit$(params, a_{i,j}, com_{i,j}, decom_{i,j}) = 1$, this algorithm outputs a secret key SK_U^i for U and an empty bit empty for $\breve{A}_i$; otherwise, it outputs $(\perp, \perp)$ to indicate that there are error messages.

F. Security Model of Privacy-Preserving Decentralized Ciphertext-Policy Attribute-Based Encryption

Informally, the security of a PPDCP-ABE scheme can be defined by any IND-sAS-CPA-secure DCP-ABE scheme with a privacy-preserving key extract algorithm PPKeyGen that satisfies two properties: *leak-freeness* and *selective-failure blindness*. Leak-freeness means that by executing the algorithm PPKeyGen with honest authorities, a malicious user cannot know anything which he cannot know by executing the algorithm KeyGen with the authorities. Selective-failure blindness means that malicious authorities cannot know anything about the user's identifier and his attributes, and cause the PPKeyGen algorithm to selectively fail depending on the user's identifier and his attributes. The following games are used to formalize these two properties.

Leak-Freeness. A real world experiment and an ideal world experiment are used to define this game.

Real World Experiment. Runs the Global Setup algorithm and Authority Setup algorithm. As many as the distinguisher $\mathcal{D}$ wants, the malicious user $\mathcal{U}$ selects a global identifier $GID_{\mathcal{U}}$ and a set of attributes $\tilde{\mathcal{U}}$, and executes PPKeyGen$(U(params, GID_{\mathcal{U}}, \tilde{\mathcal{U}}, PK_i, decom_i, (decom_{i,j})_{a_{i,j}\in\tilde{\mathcal{U}}\bigcap\tilde{A}_i}) \leftrightarrow \breve{A}_i(params, SK_i, PK_i, com_i, (com_{i,j})_{a_{i,j}\in\bigcap\tilde{\mathcal{U}}\bigcap\tilde{A}_i})) \rightarrow (SK_{\mathcal{U}}^i, \text{empty})$ with $\breve{A}_i$.

Ideal World Experiment. Runs the Global Setup algorithm and Authority Setup algorithm. As many as the distinguisher $\mathcal{D}$ wants, the malicious user $\bar{U}$ selects a global identifier $GID_{\bar{U}}$ and a set of attributes $\tilde{\bar{U}}$, and requires a trusted party to obtain the output of KeyGen$(params, SK_i, GID_{\bar{U}}, \tilde{\bar{U}}\bigcap\tilde{A}_i) \rightarrow SK_{\bar{U}}^i$.

Definition 7. *We say that an algorithm* PPKeyGen$(U \leftrightarrow \breve{A}_i)$ *associated with a DCP-ABE scheme* $\prod =$ (GlobalSetup, AuthoritySetup, Encrypt, KeyGen, Decrypt) *is leak-free if for all efficient adversary* $\mathcal{U}$*, there exists a simulator* $\bar{U}$ *such that, for the security parameter* 1^κ*, no distinguisher* $\mathcal{D}$ *can distinguish whether* $\mathcal{U}$ *is playing in the real world experiment or in the ideal world experiment with non-negligible advantage.*

Selective-Failure Blindness. This game is formally defined as follows.

1) The malicious authority $\mathcal{A}_i$ outputs his public key PK_i and two pairs of globe identifiers and attribute sets $(GID_{U_0}, \tilde{U}_0)$ and $(GID_{U_1}, \tilde{U}_1)$.
2) A random bit $b \in \{0,1\}$ is choosen.
3) $\mathcal{A}_i$ is given comments

$$\left\{com_b, (com_{i,j})_{a_{i,j}\in\tilde{U}_b\bigcap\tilde{\mathcal{A}}_i}\right\}$$

and

$$\left\{com_{1-b}, (com_{i,j})_{a_{i,j}\in\tilde{U}_{1-b}\bigcap\tilde{\mathcal{A}}_i}\right\},$$

and can black-box access oracles $U(params, GID_{U_b}, \tilde{U}_b, PK_i, decom_b, (decom_{i,j})_{a_{i,j}\in\tilde{U}_b\bigcap\tilde{A}_i})$ and $U(params, GID_{U_{1-b}}, \tilde{U}_{1-b}, PK_i, decom_{1-b}, (decom_{i,j})_{a_{i,j}\in\tilde{U}_{1-b}\bigcap\tilde{\mathcal{A}}_i})$.
4) The algorithm U outputs the secret keys $SK_{U_b}^i$ and $SK_{U_{1-b}}^i$, respectively.
5) If $SK_{U_b}^i \neq\perp$ and $SK_{U_{1-b}}^i \neq\perp$, $\mathcal{A}_i$ is given $(SK_{U_b}^i, SK_{U_{1-b}}^i)$; if $SK_{U_b}^i \neq\perp$ and $SK_{U_{1-b}}^i =\perp$, $\mathcal{A}_i$

is given $(\epsilon, \perp)$; if $SK^i_{U_b} = \perp$ and $SK^i_{U_{1-b}} \neq \perp$, $\mathcal{A}_i$ is given $(\perp, \epsilon)$; if $SK^i_{U_b} = \perp$ and $SK^i_{U_{1-b}} = \perp$, $\mathcal{A}_i$ is given $(\perp, \perp)$.

6) Finally, $\mathcal{A}_i$ outputs his guess b' on b. $\mathcal{A}_i$ wins the game if $b' = b$.

Definition 8. *We say that an algorithm* PPKeyGen$(U \leftrightarrow \breve{A}_i)$ *associated to a DCP-ABE scheme* $\prod =$ (Global Setup, Authority Setup, Encrypt, KeyGen, Decrypt) *is selective-failure blind if no probably polynomial-time adversary $\mathcal{A}_i$ can win the above game with the advantage*

$$Adv^{SFB}_{\mathcal{A}_i} = \left| \Pr[b' = b] - \frac{1}{2} \right| > \epsilon(\kappa),$$

where the probability is taken over the bits consumed by all the algorithms and the adversary.

Definition 9. *We say that a privacy-preserving decentralized ciphertext-policy attribute-based encryption (PPDCP-ABE) scheme* $\prod =$ (Global Setup, Authority Setup, Encrypt, PPKeyGen, Decrypt) *is secure if and only if the following conditions can be satisfied:*

1) $\prod =$ (Global Setup, Authority Setup, Encrypt, KeyGen, Decrypt) *is a secure DCP-ABE in the selective-access structures model;*
2) *the* PPKeyGen *algorithm is both leak-free and selective-failure blind.*

IV. Our Constructions

In this section, A PPDCP-ABE scheme is proposed.

A. DCP-ABE: Decentralized Ciphertext-policy Attribute-Based Encryption

High-Level Overview. Suppose that there are N authorities $\{\breve{A}_1, \breve{A}_2, \cdots, \breve{A}_N\}$ in the scheme, and each authority $\breve{A}_i$ monitors a set of attributes $\tilde{A}_i$ for $i = 1, 2, \cdots, N$. First, each $\breve{A}_i$ generates his secret-public key pair $\mathcal{KG}(1^\kappa) \rightarrow (SK_i, PK_i)$. For each attribute $a_{i,j} \in \tilde{A}_i$, $\breve{A}_i$ selects a random number $z_{i,j} \xleftarrow{\$} \mathbb{Z}_p$. Then, the public key and the unforgeable authentication tag are computed as $Z_{i,j} = g^{z_{i,j}}$ and $T_{i,j} = h^{z_{i,j}} g^{\frac{1}{\gamma_i + a_{i,j}}}$, respectively, where γ_i is the partial secret key of $\breve{A}_i$. As a result, $T_{i,j}$ can be used by a user to convince $\breve{A}_i$ that the attribute $a_{i,j}$ is monitored by him without releasing it. $(Z_{i,j}, T_{i,j})_{a_{i,j} \in \tilde{A}_i}$ are included in the public key PK_i.

To encrypt a message $\mathcal{M}$ under the attributes monitored by the authorities $\{\breve{A}_j\}_{j \in \mathcal{I}}$, the encryptor chooses a random number $s_j \xleftarrow{\$} \mathbb{Z}_p$ and an access structure (M_j, ρ_j) for each $\breve{A}_j$. Then, s_j is split into shares $\lambda_{j,i}$ according to the LSSS technique. Finally, the message $\mathcal{M}$ is blinded with $\prod_{j \in \mathcal{I}} e(g, g)^{\alpha_j s_j}$.

In order to resist the collusion attacks, when creating a secret key for a user U with GID μ and a set of attributes $\tilde{U}$, $\breve{A}_i$ selects two random numbers $(t_{U,i}, w_{U,i}) \xleftarrow{\$} \mathbb{Z}_p$. In details, $t_{U,i}$ is used to tie the user's attribute keys to his GID by computing $\mathfrak{g}^{t_{U,i}} \mathfrak{g}^{\frac{\beta_i + \mu}{t_{U,i}}}$ where β_i is the partial secret key of $\breve{A}_i$, and $w_{U,i}$ is used to randomize the public keys by computing $(F_x = Z_x^{w_{U,i}})_{a_x \in \tilde{U} \bigcap \tilde{A}_i}$. Then, $\breve{A}_i$ can generate a secret key for U by using his secret key and $(t_{U,i}, w_{U,i})$.

To decrypt a ciphertext, each $e(g, g)^{\alpha_j s_j}$ must be reconstructed. If the attributes in $\tilde{U}$ satisfy the access structures $(M_j, \rho_j)_{j \in \mathcal{I}}$, the user can use his secret keys and the corresponding ciphertexte elements to reconstruct $e(g, g)^{\alpha_j s_j}$, and obtain $\mathcal{M}$.

Our DCP-ABE scheme is formally described in Fig.1.

Correctness. The scheme described in Fig. 1 is correct as the following equations hold.

$$\prod_{j \in \mathcal{I}} e(K_j, X_j) = \prod_{j \in \mathcal{I}} e(g^{\alpha_j} g^{x_j w_{U,j}} \mathfrak{g}^{t_{U,j}} \mathfrak{g}^{\frac{\beta_j + \mu}{t_{U,j}}}, g^{s_j}) =$$
$$\prod_{j \in \mathcal{I}} e(g, g)^{\alpha_j s_j} \cdot e(g, g)^{x_j w_{U,j} s_j} \cdot e(g, \mathfrak{g})^{t_{U,j} s_j} \cdot e(g, \mathfrak{g})^{\frac{\beta_j s_j}{t_{U,j}}} \cdot e(g, \mathfrak{g})^{\frac{\mu s_j}{t_{U,j}}},$$

$$\prod_{j \in \mathcal{I}} e(R_j, E_j) \cdot e(R_j, Y_j)^\mu = \prod_{j \in \mathcal{I}} e(g^{\frac{1}{t_{U,j}}}, B_j^{s_j}) \cdot e(g^{\frac{1}{t_{U,j}}}, \mathfrak{g}^{s_j})^\mu$$
$$= \prod_{j \in \mathcal{I}} e(g^{\frac{1}{t_{U,j}}}, \mathfrak{g}^{\beta_j s_j}) \cdot e(g^{\frac{1}{t_{U,j}}}, \mathfrak{g}^{s_j})^\mu$$
$$= \prod_{j \in \mathcal{I}} e(g, \mathfrak{g})^{\frac{\beta_j s_j}{t_{U,j}}} \cdot e(g, \mathfrak{g})^{\frac{\mu s_j}{t_{U,j}}},$$

$$\prod_{j \in \mathcal{I}} e(L_j, X_j) = e(g, g)^{t_{U,j} s_j},$$

and

$$\prod_{j \in \mathcal{I}} \prod_{i=1}^{\ell_j} \left(e(C_{j,i}, P_j) \cdot e(D_{j,i}, F_{\rho_j(i)}) \right)^{\omega_{j,i}}$$
$$= \prod_{j \in \mathcal{I}} \prod_{i=1}^{\ell_j} \left(e(g^{g^{x_j \lambda_{j,i}}} Z_{\rho_j(i)}^{-r_{j,i}}, g^{w_{U,j}}) \cdot e(g^{r_{j,i}}, Z_{\rho_j(i)}^{w_{U,j}}) \right)^{\omega_{j,i}}$$
$$= \prod_{j \in \mathcal{I}} e(g, g)^{x_j w_{U,j} \sum_{i=1}^{\ell_j} \omega_{j,i} \lambda_{j,i}}$$
$$= \prod_{j \in \mathcal{I}} e(g, g)^{x_j w_{U,j} s_j}.$$

Therefore,

$$\frac{C_0 \cdot \prod_{j \in \mathcal{I}} e(L_j, X_j) \cdot e(R_j, E_j) \cdot e(R_j, Y_j)^\mu}{\prod_{j \in \mathcal{I}} e(K_j, X_j)} \cdot \prod_{j \in \mathcal{I}} \prod_{i=1}^{\ell_j} \left(e(C_{j,i}, P_j) \cdot e(D_{j,i}, F_{\rho_j(i)}) \right)^{\omega_{j,i}} = \mathcal{M}.$$

B. Security of the Proposed DCP-ABE

Theorem 2. *Our decentralized ciphertext-policy attribute-based encryption (DCP-ABE) in Fig. 1 is $(T, q, \epsilon(k))$ secure in the selective-access structure model if the $(T', \epsilon'(k))$-decisional q-PBDHE assumption holds on $(e, p.\mathbb{G}, \mathbb{G}_\tau)$, where $T' = T + \mathcal{O}(T)$ and $\epsilon'(\kappa) = \frac{1}{2}\epsilon(\kappa)$.*

Proof: Suppose that there exists an adversary $\mathcal{A}$ who can $(T, q, \epsilon(k))$ break our DCP-ABE in Fig. 1, we will show that

Global Setup. Taking as input a security parameter 1^κ, this algorithm outputs a bilinear group $\mathcal{GG}(1^\kappa) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$. Let g, h and $\mathfrak{g}$ be generators of the group $\mathbb{G}$. Suppose that there are N authorities $\{\breve{A}_1, \breve{A}_2, \cdots, \breve{A}_N\}$, and $\breve{A}_i$ monitors a set of attributes $\tilde{A}_i = \{a_{i,1}, a_{i,2}, \cdots, a_{i,q_i}\}$ where $a_{i,j} \in \mathbb{Z}_p$ for $i = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, q_i$. The public parameters are $PP = (g, h, \mathfrak{g}, e, p, \mathbb{G}, \mathbb{G}_\tau)$.

Authorities Setup. Each authority $\breve{A}_i$ chooses $\alpha_i, x_i, \beta_i, \gamma_i \xleftarrow{\$} \mathbb{Z}_p$, and computes $H_i = e(g, g)^{\alpha_i}$, $A_i = g^{x_i}$, $B_i = \mathfrak{g}^{\beta_i}$, $\Gamma_i^1 = g^{\gamma_i}$ and $\Gamma_i^2 = h^{\gamma_i}$, where $i = 1, 2, \cdots, N$. For each attribute $a_{i,j} \in \tilde{A}_i$, $\breve{A}$ selects $z_{i,j} \xleftarrow{\$} \mathbb{Z}_p$, and computes $Z_{i,j} = g^{z_{i,j}}$ and $T_{i,j} = h^{z_{i,j}} g^{\frac{1}{\gamma_i + a_{i,j}}}$. Then, $\breve{A}$ publishes the public key $PK_i = \left\{H_i, A_i, B_i, (\Gamma_i^1, \Gamma_i^2), (T_{i,j}, Z_{i,j})_{a_{i,j} \in \tilde{A}_i}\right\}$, and keeps the master secrete key $SK_i = (\alpha_i, a_i, \beta_i, \gamma_i, (z_{i,j})_{a_{i,j} \in \tilde{A}_i})$ private.

Encryption. To encrypt a message $\mathcal{M} \in \mathbb{G}_\tau$, this algorithm works as follows. Let $\mathcal{I}$ be a set which consists of the indexes of the authorities whose attributes are selected to encrypt $\mathcal{M}$. For each $j \in \mathcal{I}$, this algorithm first chooses an access structures (M_j, ρ_j) and a vector $\overrightarrow{v_j} = (s_j, v_{j,2}, \cdots, v_{j,n_j})$, where $s_j, v_{j,2}, \cdots, v_{j,n_j} \xleftarrow{\$} \mathbb{Z}_p$ and M_j is an $\ell_j \times n_j$ matrix. Then, it computes $\lambda_{j,i} = M_j^i \overrightarrow{v}_j$, where M_j^i is the corresponding ith row of M_j. Finally, it selects $r_{j,1}, r_{j,2}, \cdots, r_{j,\ell_j} \xleftarrow{\$} \mathbb{Z}_p$, and computes

$$C_0 = \mathcal{M} \cdot \prod_{j \in \mathcal{I}} e(g, g)^{\alpha_j s_j}, \ \{X_j = g^{s_j}, \ Y_j = \mathfrak{g}^{s_j}, \ E_j = B_j^{s_j}\}_{j \in \mathcal{I}}$$

$$\left((C_{j,1} = g^{x_j \lambda_{j,1}} Z_{\rho_j(1)}^{-r_{j,1}}, \ D_{j,1} = g^{r_{j,1}}), \ \cdots, \ (C_{j,\ell_j} = g^{x_j \lambda_{j,\ell_j}} Z_{\rho_j(\ell_j)}^{-r_{j,\ell_j}}, \ D_{j,\ell_j} = g^{r_{j,\ell_j}})\right)_{j \in \mathcal{I}}$$

The ciphertext is $CT = \left\{C_0, \ \left(X_j, \ Y_j, \ E_j, \ (C_{j,1}, \ D_{j,1}), \ \cdots, \ (C_{j,\ell_j}, \ D_{j,\ell_j})\right)_{j \in \mathcal{I}}\right\}$.

KeyGen. To generate secret keys for a user U with GID μ and a set of attributes $\tilde{U} \bigcap \tilde{A}_i$, $\breve{A}_i$ chooses $t_{U,i}, w_{U,i} \xleftarrow{\$} \mathbb{Z}_p$, and computes $K_i = g^{\alpha_i} g^{x_i w_{U,i}} \mathfrak{g}^{t_{U,i}} \mathfrak{g}^{\frac{\beta_i + \mu}{t_{U,i}}}$, $P_i = g^{w_{U,i}}$, $L_i = g^{t_{U,i}}$, $L_i' = h^{t_{U,i}}$, $R_i = g^{\frac{1}{t_{U,i}}}$, $R_i' = h^{\frac{1}{t_{U,i}}}$ and $(F_x = Z_x^{w_{U,i}})_{a_x \in \tilde{U} \bigcap \tilde{A}_i}$.

The secret keys for U are $SK_U^i = \left\{K_i, P_i, L_i, L_i', R_i, R_i', (F_x)_{a_x \in \tilde{U} \bigcap \tilde{A}_i}\right\}$.

Decryption. To decrypt a ciphertext CT, this algorithm computes

$$\frac{C_0 \cdot \prod_{j \in \mathcal{I}} e(L_j, X_j) \cdot e(R_j, E_j) \cdot e(R_j, Y_j)^\mu \cdot \prod_{j \in \mathcal{I}} \prod_{i=1}^{\ell_j} \left(e(C_{j,i}, P_j) \cdot e(D_{j,i}, F_{\rho_j(i)})\right)^{\omega_{j,i}}}{\prod_{j \in \mathcal{I}} e(K_j, X_j)} = \mathcal{M}$$

where $\{\omega_{j,i} \in \mathbb{Z}_p\}_{i=1}^{\ell_j}$ are a set of constants such that $\sum_{i=1}^{\ell_j} \omega_{j,i} \lambda_{j,i} = s_j$ if $\{\lambda_{j,i}\}_{i=1}^{\ell_j}$ are valid shares of the secret value s_j according to the access structure (M_j, ρ_j).

Fig. 1: DCP-ABE: Decentralized Ciphertext-Policy Attribute-based Encryption

there exists an algorithm $\mathcal{B}$ which can use $\mathcal{A}$ to break the decisional q-PDHE assumption as follows.

The challenger generates the bilinear group $\mathcal{GG}(1^k) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$, and chooses a generators $g \in \mathbb{G}$. Let $\overrightarrow{y} =$

$$g, g^s, g^a, \cdots, g^{(a^q)}, g^{(a^{q+3})}, \cdots, g^{(a^{2q})}$$

$$\forall_{1 \leq j \leq q} \quad g^{s \cdot b_j}, g^{\frac{a}{b_j}}, \cdots, g^{(\frac{a^q}{b_j})}, g^{(\frac{a^{q+2}}{b_j})}, \cdots, g^{(\frac{a^{2q}}{b_j})}$$

$$\forall_{1 \leq j,k \leq q, k \neq j} \quad g^{\frac{a \cdot s \cdot b_k}{b_j}}, \cdots, g^{(\frac{a^q \cdot s \cdot b_k}{b_j})}.$$

The challenger flips an unbiased coin with $\{0, 1\}$, and obtains a bit $\vartheta \in \{0, 1\}$. If $\vartheta = 0$, he sends $(\overrightarrow{y}, \Omega = e(g, g)^{a^{q+1} s})$ to $\mathcal{B}$; otherwise, he sends $(\overrightarrow{y}, \Omega = V)$ to $\mathcal{B}$ where $V \xleftarrow{\$} \mathbb{G}_\tau$. $\mathcal{B}$ will output his guess ϑ' on ϑ.

Initialization. The adversary $\mathcal{A}$ submits a list of corrupted authorities with index $\mathcal{I}'$ and challenge access structures $\mathbb{A} = \left\{(M_j^*, \rho_j^*)\right\}_{j \in \mathcal{I}^*}$ where $\mathcal{I}^*$ is a set consisting of the indexes of the authorities $\breve{A}_j$. Let M^* be a $\ell^* \times n^*$ matrix and $\ell^*, n^* < q$. Suppose that (M^*, ρ^*) is specified by the authority $\breve{A}^*$ with $\breve{A}^* \notin \mathbb{A}$ and cannot be satisfied by the attributes selected by $\mathcal{A}$ to query secrete keys.

Globe Setup. $\mathcal{B}$ selects $\pi, \varrho \xleftarrow{\$} \mathbb{Z}_p$, and computes $h = g^\pi$ and $\mathfrak{g} = g^\varrho$. Then, $\mathcal{B}$ sends $PP = (g, \mathfrak{g}, h, e, p, \mathbb{G}, \mathbb{G}_\tau)$ to $\mathcal{A}$.

Authorities Setup.

1) For the authority $\breve{A}_i$ with $i \in \mathcal{I}'$, $\mathcal{B}$ chooses $\alpha_i, x_i, \beta_i, \gamma_i, z_{i,j} \xleftarrow{\$} \mathbb{Z}_p$, and sets $Y_i = e(g, g)^{\alpha_i}$, $A_i = g^{b_i}$, $B_i = \mathfrak{g}^{\beta_i}$, $\Gamma_i^1 = g^{\gamma_i}, \Gamma_i^2 = h^{\gamma_i}$ and $\left(Z_{i,j} = g^{z_{i_j}}, \ T_{i,j} = Z_{i,j}^\pi g^{\frac{1}{\gamma_i + a_{i,j}}}\right)_{a_{i,j} \in \tilde{A}}$. This implies that the master secret key of $\breve{A}_i$ is $SK_i = \left(\alpha_i, x_i, \beta_i, \gamma_i, (z_{i,j})_{a_{i,j} \in \tilde{A}_i}\right)$ and the public key is $PK_i = \left(Y_i, A_i, B_i, \Gamma_i^1, \Gamma_i^2, (T_{i,j}, Z_{i,j})_{a_{i,j} \in \tilde{A}_i}\right)$. $\mathcal{B}$ sends (Sk_i, Pk_i) to $\mathcal{A}$.
2) For the authority $\breve{A}_i$ with $i \notin \mathcal{I}'$ and $\breve{A}_i \neq \breve{A}^*$, it

chooses $\alpha_i, x_i, \beta_i, \gamma_i, z_{i,j} \xleftarrow{\$} \mathbb{Z}_p$, and computes $Y_i = e(g,g)^{\alpha_i}$, $A_i = g^{x_i}$, $B_i = \mathfrak{g}^{\beta_i}$, $\Gamma_i^1 = g^{\gamma_i}$, $\Gamma_i^2 = h^{\gamma_i}$ and $\left(Z_{i,j} = g^{z_{i_j}},\ T_{i,j} = Z_{i,j}^{\pi} g^{\frac{1}{\gamma_i + a_{i,j}}}\right)_{j=1}^{n_i}$. This implies that the master secret key of $\breve{A}_i$ is $SK_i = \left(\alpha_i, x_i, \beta_i, \gamma_i, (z_{i,j})_{j=1}^{n_i}\right)$ and the public key is $PK_i = \left(Y_i, A_i, B_i, \Gamma_i^1, \Gamma_i^2, (T_{i,j}, Z_{i,j})_{j=1}^{n_i}\right)$. $\mathcal{B}$ sends PK_i to $\mathcal{A}$.

3) For the authority $\breve{A}^*$, $\mathcal{B}$ chooses $\alpha', \beta, \gamma \xleftarrow{\$} \mathbb{Z}_p$, sets $\alpha = \alpha' + a^{q+1} + \sum_{i \in \mathcal{I}'} \alpha_i$, and computes

$$Y^* = e(g,g)^{\alpha} = e(g^{\alpha}, g^{\alpha^q}) \cdot e(g,g)^{\alpha'} \prod_{i \in \mathcal{I}'} e(g,g)^{-\alpha_i},$$

$$A^* = g^a,\ B^* = \mathfrak{g}^{\beta},\ \Gamma^{*1} = g^{\gamma},\ \Gamma^{*2} = h^{\gamma}.$$

Let $\mathcal{X}$ be the set consisting of the indexes i with $\rho^*(i) = x$ for $i = 1, 2, \cdots, \ell^*$.

   a) For the attribute a_x with $\rho^*(i) = x$, $\mathcal{B}$ chooses $z_x \xleftarrow{\$} \mathbb{Z}_p$ and computes $Z_x = g^{z_x} \prod_{i \in X} g^{\frac{aM^*_{i,1}}{b_i}} \cdot g^{\frac{a^2 M^*_{i,2}}{b_i}} \cdots g^{\frac{a^{n^*} M^*_{i,n^*}}{b_i}}$ and $T_x = Z_x^{\pi} g^{\frac{!}{\gamma + a_x}}$.
   b) For the attributes a_x with $\rho^*(i) \neq x$, $\mathcal{B}$ chooses $z_x \xleftarrow{\$} \mathbb{Z}_p$, and computes $Z_x = g^{z_x}$ and $T_x = h^{z_x} g^{\frac{1}{\gamma + a_x}}$.

This implies that the master secrete key of $\breve{A}^*$ is $SK^* = (\alpha, a, b, \gamma, (z_x + \sum_{i \in X}(\frac{aM^*_{i,1}}{b_i} + \cdots + \frac{a^{n^*} M^*_{i,n^*}}{b_i}))_{\rho^*(i)=x}, (z_x)_{\rho^*(i) \neq x}))$ and the public key is $PK^* = (Y^*, A^*, B^*, (T_x, Z_x)_{a_x \in \tilde{A}^*})$. Then, $\mathcal{B}$ sends PK^* to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ can adaptively query secrete key for a user U with a globe identifier μ and a set of attribute $\tilde{U}$ which does not satisfy M^*. $\mathcal{B}$ works as follows.

1) For the authority $\breve{A}_i$ with $i \in \mathcal{I}'$, $\mathcal{B}$ chooses $w_i, t_i \xleftarrow{\$} \mathbb{Z}_p$, and computes $K_i = g^{\alpha_i} g^{x_i w_i} \mathfrak{g}^{t_i} \mathfrak{g}^{\frac{\beta_i + \mu}{t_i}}$, $P_i = g^{w_i}$, $L_i = g^{t_i}$, $L_i' = L_i^{\pi}$, $R_i = \mathfrak{g}^{\frac{1}{t_i}}$, $R_i' = R_i^{\pi}$ and $(F_x = T_x^{w_i})_{a_{x \in \tilde{A}_i \cap \tilde{U}}}$. $\mathcal{B}$ sends the secret key $SK_U^i = \left\{K_i, P_i, L_i, L_i', R_i, (F_x)_{a_{x \in \tilde{A}_i \cap \tilde{U}}}\right\}$ to $\mathcal{A}$.
2) For the authority $\breve{A}_i$ with $i \notin \mathcal{I}'$ and $\breve{A}_i \neq \breve{A}^*$, $\mathcal{B}$ chooses $w_i, t_i \xleftarrow{\$} \mathbb{Z}_p$, and computes $K_i = g^{\alpha_i} g^{x_i w_i} \mathfrak{g}^{t_i} \mathfrak{g}^{\frac{\beta_i + \mu}{t_i}}$, $P_i = g^{w_i}$, $L_i = g^{t_i}$, $L_i' = L_i^{\pi}$, $R_i = g^{\frac{1}{t_i}}$, $Ri' = R_i^{\pi}$ and $(F_x = T_x^{w_i})_{a_{x \in \tilde{A}_i \cap \tilde{U}}}$. $\mathcal{B}$ sends the secret key $SK_U^i = \left\{K_i, P_i, L_i, L_i, R_i, (F_x)_{a_{x \in \tilde{A}_i \cap \tilde{U}}}\right\}$ to $\mathcal{A}$.
3) For the authority $\breve{A}^*$, $\mathcal{B}$ chooses $t, r \xleftarrow{\$} \mathbb{Z}_p$ and a vector $\overrightarrow{f} = (f_1, f_2, \cdots, f_{n^*}) \in \mathbb{Z}_p^{n^*}$ such that $f_1 = -1$ and $\overrightarrow{f} \cdot M_i^* = 0$ for all $\rho^*(i) \in \tilde{U} \bigcap \tilde{A}^*$. It computes $P = g^r \prod_{i=1}^{n^*} g^{f_i a^{q-i+1}} = g^w$. By this, $\mathcal{B}$ implicitly defines $w = r + f_1 a^q + f_2 a^{q-1} + \cdots + f_{n^*} a^{q-n^*+1}$. Then, $\mathcal{B}$ computes $K = g^{\alpha' - \sum_{i \in \mathcal{I}'} \alpha_i} g^{ra} \prod_{i=2}^{n^*} g^{f_i a^{q-i+2}} \mathfrak{g}^t \mathfrak{g}^{\frac{\beta + \mu}{t}}$, $L = g^t$, $L' = L^{\pi}$, $R = g^{\frac{1}{t}}$ and $R' = R^{\pi}$.
   a) For the attribute $a_x \in \tilde{A}^* \bigcap \tilde{U}$ for which there is no i such that $\rho^*(i) = x$, $\mathcal{B}$ computes $F_x = P^{z_x}$
   b) For the attributes $a_x \in \tilde{A}^* \bigcap \tilde{U}$ for which there does exist an i such that $\rho^*(i) = x$, $\mathcal{B}$ computes $(F_x = P^{z_x} \prod_{i \in X} \prod_{j=1}^{n^*} (g^{\frac{ra^j}{b_i}} \prod_{k=1, k \neq j}^{n^*} g^{\frac{f_k a^{q+1+j-k}}{b_i}})^{M^*_{i,j}})$.

   $\mathcal{B}$ sends the secret key $SK = (K, P, L, L', R, R', (F_x)_{a_x \in \tilde{U} \bigcap \tilde{A}^*})$ to $\mathcal{A}$

We claim that the secret key created above are correct as we have

$K = g^{\alpha' - \sum_{i \in \mathcal{I}'} \alpha_i} g^{ra} \prod_{i=2}^{n^*} g^{f_i a^{q-i+2}} \mathfrak{g}^t \mathfrak{g}^{\frac{\beta + \mu}{t}} = g^{\alpha} g^{ra} \prod_{i=1}^{n^*} g^{f_i a^{q-i+2}} \mathfrak{g}^t \mathfrak{g}^{\frac{\beta + \mu}{t}} = g^{\alpha} g^{a(r + \sum_{i=1}^{n^*} f_i a^{q-i+1})} \mathfrak{g}^t \mathfrak{g}^{\frac{\beta + \mu}{t}} = g^{\alpha} g^{aw} \mathfrak{g}^t \mathfrak{g}^{\frac{\beta + \mu}{t}}$,

$P = g^r \prod_{i=1}^{n^*} g^{f_i a^{q-i+1}} = g^{r + \sum_{i=1}^{n^*} f_i a^{q-i+1}} = g^w$,

$L = g^t$, $L' = L^{\pi} = h^t$, $R = g^{\frac{1}{t}}$ and $R' = R^{\pi} = h^{\frac{1}{t}}$.

For the attribute $a_x \in \breve{A}^* \bigcap \tilde{U}$ for which there is no an i such that $\rho^*(i) = x$, $F_x = P^{z_x} = (g^w)^{z_x} = (g^{z_x})^w = Z_x^w$.

For the attribute $a_x \in \breve{A}^* \bigcap \tilde{U}$ for which there does exist an i such that $\rho^*(i) = x$,

$$\begin{aligned}
F_x &= P^{z_x} \prod_{i \in X} \prod_{j=1}^{n^*} \left(g^{\frac{ra^j}{b_i}} \prod_{k=1, k \neq j}^{n^*} g^{\frac{f_k a^{q+1+j-k}}{b_i}} \right)^{M^*_{i,j}} \\
&= (g^{z_x})^w \prod_{i \in X} \prod_{j=1}^{n^*} \left(g^{\frac{ra^j}{b_i}} \prod_{k=1}^{n^*} g^{\frac{f_k a^{q+1+j-k}}{b_i}} \right)^{M^*_{i,j}} \\
&= (g^{z_x})^w \prod_{i \in X} g^{\sum_{j=1}^{n^*} \frac{ra^j M^*_{i,j}}{b_i}} \cdot g^{\sum_{k=1}^{n^*} f_k a^{q-k+1} \sum_{j=1}^{n^*} \frac{M^*_{i,j} a^j}{b_i}} \\
&= (g^{z_x})^w \prod_{i \in X} g^{\sum_{k=1}^{n^*} (r + f_k a^{q-k+1}) \sum_{j=1}^{n^*} \frac{M^*_{i,j} a^j}{b_i}} \\
&= (g^{z_x})^w \prod_{i \in X} g^{w \sum_{j=1}^{n^*} \frac{M^*_{i,j} a^j}{b_i}} \\
&= \left(g^{z_x} \prod_{i \in X} g^{\frac{M^*_{i,1} a}{b_i}} g^{\frac{M^*_{i,2} a^2}{b_i}} \cdots g^{\frac{M^*_{i,n^*} a^{n^*}}{b_i}} \right)^w \\
&= Z_x^w
\end{aligned}$$

Challenge. $\mathcal{A}$ submits two messages $\mathcal{M}_0$ and $\mathcal{M}_1$ with the same length to $\mathcal{B}$. $\mathcal{B}$ flips an unbiased coin with $\{0, 1\}$, and obtains a bit ϑ.

1) For the authority $\breve{A}_i$ with $i \in \mathcal{I}^*$ and $\breve{A}_i \neq \breve{A}^*$, $\mathcal{B}$ chooses $s_i \xleftarrow{\$} \mathbb{Z}_p$ and computes $X_i = g^s g^{-s_i}$, $Y_i = X_i^{\varrho}$, $E_i = (g^s g^{-s_i})^{\varrho \beta_i}$. Then, $\mathcal{B}$ chooses $r_{i,1}, r_{i,2}, \cdots, r_{i,\ell_i}, v_{i,2}, v_{i,3}, \cdots, v_{i,n_i} \xleftarrow{\$} \mathbb{Z}_p$, and sets $\overrightarrow{v}_i = (s - s_i, v_{i,2}, \cdots, v_{i,n_i})$ which is used to share the secrete $(-s_i)$. $\mathcal{B}$ computes $C_{i,k} = g^{sM_i^{k,1}} g^{-s_i} \prod_{j=2}^{n_i} g^{v_{i,j} M_i^{k,j}} Z_{\rho_i(k)}^{-r_{i,k}}$ and $D_{i,k} = g^{r_{i,k}}$ where $k = 1, 2, \cdots, \ell_i$ and $M_i^{k,j}$ denotes the element in the position (k, j) of the matrix M_i.
2) For the authority $\breve{A}^*$, $\mathcal{B}$ computes $X = g^s$, $Y = g^{s\varrho}$, $E = g^{s\beta\varrho}$. Then, $\mathcal{B}$ chooses $r_1, r_2, \cdots, r_{n^*}, v_2, v_3, \cdots, v_{n^*} \xleftarrow{\$} \mathbb{Z}_p$, and sets $\overrightarrow{v} = (s, sa + v_2, sa^2 + v_3, \cdots, sa^{n^*-1} + v_{n^*})$ which is used

IEEE Transactions on Information Forensics and Security,(Volume:10 , Issue: 3),March 2015.

to share the secret s. Let $\mathcal{R}$ be a set consisting of all $i \neq j$ with $\rho^*(i) = \rho^*(j)$. $\mathcal{B}$ computes

$$C_k = Z^{r_k}_{\rho^*(k)} (\prod_{j=1}^{n^*} (g^a)^{M^*_{i,j} v_j} (g^{b_k s})^{-z_{\rho^*(k)}} \cdot (\prod_{l \in \mathcal{R}} \prod_{j=1}^{n^*} (g^{a^j s (b_k / b_l)})^{M^*_{l,j}}))$$

and $D_k = g^{-r_k} g^{-s b_k}$ where $k = 1, 2, \cdots, \ell^*$.

Finally, $\mathcal{B}$ computes $C^*_0 = M_{\hat{\vartheta}} \cdot \Omega \cdot e(g^{\alpha'}, g^s) \cdot \prod_{i \in \mathcal{I}^*, \breve{A}_i \neq \breve{A}^*} e(g,g)^{\alpha_i s}$.

The challenge ciphertext is $CT^* = (C_0, (X_j, Y_j, E_j, (C_{j,1}, D_{j,1}), \cdots, (C_{j,\ell_j}, D_{j,\ell_j}))_{i \in \mathcal{I}^*, \breve{A}_j \neq \breve{A}^*}, (X, Y, E, (C_k, D_k)^{\ell^*}_{k=1}))$.

Phase 2. Phase 1 is repeated.

Guess. $\mathcal{A}$ outputs his guess $\tilde{\vartheta}$ on $\hat{\vartheta}$. If $\tilde{\vartheta} = \hat{\vartheta}$, $\mathcal{B}$ outputs $\vartheta' = 0$; otherwise, $\mathcal{B}$ outputs $\vartheta' = 1$. As shown above, the public parameters, the public keys and secret keys created in the simulation are identical to those in the real protocol. The remaining thing is to compute the probability with which $\mathcal{B}$ can break the decisional q-PBDHE assumption.

If $\vartheta = 0$, $\Omega = e(g,g)^{a^{q+1}s}$. Then, CT^* is a correct ciphertext of $\mathcal{M}_0$. Therefore, $\mathcal{A}$ can outputs $\tilde{\vartheta} = \hat{\vartheta}$ with the advantage at least $\epsilon(\kappa)$, namely $\Pr[\tilde{\vartheta} = \hat{\vartheta} | \vartheta = 0] > \frac{1}{2} + \epsilon(\kappa)$. Since $\mathcal{B}$ outputs $\vartheta' = 0$ when $\tilde{\vartheta} = \hat{\vartheta}$, we have $\Pr[\vartheta' = \vartheta | \vartheta = 0] > \frac{1}{2} + \epsilon(\kappa)$.

If $\vartheta = 1$, Ω is a random number in $\mathbb{G}_\tau$. Therefore $\mathcal{A}$ can outputs $\tilde{\vartheta} \neq \hat{\vartheta}$ with no advantage, namely $\Pr[\tilde{\vartheta} \neq \hat{\vartheta} | \vartheta = 1] = \frac{1}{2}$. Since $\mathcal{B}$ outputs $\vartheta' = 1$ when $\tilde{\vartheta} \neq \hat{\vartheta}$, we have $\Pr[\vartheta' = \vartheta | \vartheta = 1] = \frac{1}{2}$.

Thereafter, the advantage with which $\mathcal{B}$ can break the decisional q-PBDHE is $|\frac{1}{2}\Pr[\tilde{\vartheta} = \hat{\vartheta} | \vartheta = 0] - \frac{1}{2}\Pr[\vartheta' = \vartheta | \vartheta = 1]| > \frac{1}{2} \times \frac{1}{2} + \frac{1}{2}\epsilon(\kappa) - \frac{1}{2} \times \frac{1}{2} = \frac{1}{2}\epsilon(\kappa)$. ■

C. Efficiency of The Proposed DCP-ABE

We list the computation cost and communication cost of our PPDCP-BAE scheme in Table I and Table II, respectively. N is the number of the authorities in the scheme and $\mathcal{I}$ is a set consisting of the indexes of the authorities $\breve{A}_i$ if the attributes monitored by $\breve{A}_i$ are used to encrypt a message. $\tilde{U}$ is the set of attributes held by U. q_i stands for the number of the attributes monitored by the authorities $\breve{A}_i$. ℓ_j is denoted as the number of the rows of the matrix in the access structure (M_j, ρ_j).

D. Privacy-Preserving Key Extract Protocol

High-Level Overview. In Fig. 1, to generate a secret key for a user U, the authority $\breve{A}_i$ chooses two random numbers $(t_{U,i}, w_{U,i})$, and uses them to tie the user's secret key to his GID. If $\breve{A}_i$ records $(t_{U,i}, w_{U,i})$, he can compute $\mathfrak{g}^\mu = (\frac{K_i}{g^{\alpha_i} g^{x_i w_{U,i}} \mathfrak{g}^{t_{U,i}}})^{t_{U,i}} \mathfrak{g}^{-\beta_i}$ and $(Z_x = F_x^{\frac{1}{w_{U,i}}})_{a_x \in \tilde{U} \bigcap \tilde{A}_i}$, and know the user's GID and attributes. Therefore, to protect the privacy of the user's GID and attributes, $(t_{U,i}, w_{U,i})$ should be computed using the 2-party secure computing technique.

First, U selects $(k_1, k_2, d_1, d_2) \xleftarrow{\$} \mathbb{Z}_p$. It uses (k_1, k_2) to commit his GID and (d_1, d_2) to commit his attributes and the corresponding authentication tags. Then, U proves in zero knowledge to $\breve{A}_i$ that he knows the GID, and the attributes for which he is obtaining secret keys are monitored by $\breve{A}_i$. $\breve{A}_i$ checks the proof. If it fails, $\breve{A}_i$ aborts. Otherwise, $\breve{A}_i$ selects $(c_u, e_u) \xleftarrow{\$} \mathbb{Z}_p$ and generates a secret key for U by using his secret key, the elements from U and (c_u, e_u). Furthermore, $\breve{A}_i$ proves in zero knowledge that he knows the secret key and (c_u, e_u). Finally, U can compute his real secret key by (k_1, k_2, d_1, d_2) and the elements from $\breve{A}_i$.

Actually, by executing the 2-party secure computing protocol, U and $\breve{A}_i$ cooperatively compute $w_{U,i} = e_u d_1$ and $t_{U,i} = \frac{c_u}{k_2}$, where (d_1, k_2) are from U and (c_u, e_u) are from $\breve{A}_i$. Therefore, from the view of $\breve{A}_i$, the secret key computed by U is indistinguishable from the random elements in $\mathbb{G}$.

The privacy-preserving key extract protocol PPKeyGen is described in Fig. 2.

Correctness. Let $w = d_1 e_u$ and $t = \frac{c_u}{k_2}$. The secret keys created in Fig. 2 are correct as the following equations hold.

$$\begin{aligned} K_i &= \frac{K'_i \Upsilon^{\frac{1}{k_2}}}{\Upsilon_4^{k_1 k_2}} = \frac{g^{\alpha_i} \Theta_1^{e_u} (\Theta_4 \Theta_5)^{\frac{1}{c_u}} \mathfrak{g}^{\frac{c_u}{k_2}}}{\Upsilon_4^{k_1 k_2}} \\ &= \frac{g^{\alpha_i} A_i^{d_1 e_u} ((\mathfrak{h}^{k_1} \mathfrak{g}^\mu)^{k_2} B_i^{k_2})^{\frac{1}{c_u}} \mathfrak{g}^{\frac{c_u}{k_2}}}{\mathfrak{h}^{\frac{k_1 k_2}{c_u}}} \\ &= \frac{g^{\alpha_i} g^{x_i d_1 e_u} \mathfrak{h}^{\frac{k_1 k_2}{c_u}} \mathfrak{g}^{\frac{k_2(\beta_i + \mu)}{c_u}} \mathfrak{g}^{\frac{c_u}{k_2}}}{\mathfrak{h}^{\frac{k_1 k_2}{c_u}}} \\ &= g^{\alpha_i} g^{x_i w} \mathfrak{g}^t \mathfrak{g}^{\frac{\beta_i + \mu}{t}}, \end{aligned}$$

$$P_i = \Upsilon_6^{d_1} = g^{d_1 e_u} = g^w, \; L_i = \Upsilon_1^{\frac{1}{k_2}} = g^{\frac{c_u}{k_2}} = g^t,$$
$$R_i = \Upsilon_2^{k_2} = g^{\frac{k_2}{c_u}} = g^{\frac{1}{t}}, \; R'_i = \Upsilon_4^{k_2} = h^{\frac{k_2}{c_u}} = h^{\frac{1}{t}}$$

and

$$F_x = \Phi_x^{\frac{1}{d_2}} = (\Psi_x^2)^{\frac{e_u}{d_2}} = Z_x^{\frac{d_u e_u}{d_2}} = Z_x^{d_1 e_u} = Z_x^w.$$

E. An Instance of the PPKeyGen Protocol

The details of the protocol in Fig. 2 are as follows.

1) U selects $k_1, k_2, d_1, d_2, k'_1, k'_2, k'_3, k'_4, k'_5, k'_6, d'_1, d'_2 \xleftarrow{\$} \mathbb{Z}_p$, and sets $d_u = d_1 d_2$ and $d'_u = d'_1 d'_2$. It computes $\Theta_1 = A_i^{d_1}$, $\Theta_2 = g^{d_u}$, $\Theta_3 = h^{k_1} \mathfrak{g}^\mu$, $\Theta_4 = \Theta_3^{k_2}$, $\Theta_5 = B_i^{k_2}$, $\Theta_6 = \mathfrak{g}^{\frac{1}{k_2}}$, $(\Psi_x^1 = T_x^{d_u}, \Psi_x^2 = Z_x^{d_u})_{a_x \in \tilde{U} \bigcap \tilde{A}_i}$, $\Theta'_1 = A_i^{d'_1}$, $\Theta'_2 = g^{d'_u}$, $\Theta'_3 = h^{k'_1} \mathfrak{g}^{k'_3}$, $\Theta'_4 = \Theta_3^{k'_2}$, $\Theta'_5 = B_i^{k'_2}$, $\Theta'_6 = \mathfrak{g}^{\frac{1}{k'_2}}$ $(\Psi_x^3 = h^{k'_4} g^{a_x}$, $\Psi_x^4 = h^{k'_6} g^{k'_5}$, $\Psi_x^5 = e(h, \Psi_x^2)^{k'_5} \cdot e(g, \Psi_x^1)^{-k'_5} \cdot e(g,g)^{d'_u})_{a_x \in \tilde{U} \bigcap \tilde{A}_i}$. Then, U sends $(\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5, \Theta_6, \Theta'_1, \Theta'_2, \Theta'_3, \Theta'_4, \Theta'_5, \Theta'_6, (\Psi_x^1, \Psi_x^2, \Psi_x^3, \Psi_x^4, \Psi_x^5)_{a_x \in \tilde{U} \bigcap \tilde{A}_i})$ to $\breve{A}_i$.

2) $\breve{A}_i$ selects $\eta \xleftarrow{\$} \mathbb{Z}_p$, and sends it to U.

3) U computes $\tilde{d}_1 = d'_1 - \eta d_1$, $\tilde{d}_u = d'_u - \eta d_u$, $\tilde{k}_1 = k'_1 - \eta k_1$, $\tilde{k}_2 = k'_2 - \eta k_2$, $\tilde{k}_3 = k_3 - \eta \mu$, $\tilde{k}_4 = k'_7 - \eta k'_4$ $\tilde{k}_5 = k'_5 - \eta a_x$, and $\tilde{k}_6 = \frac{1}{k'_2} - \eta \frac{1}{k_2}$.

IEEE Transactions on Information Forensics and Security,(Volume:10 , Issue: 3),March 2015.

TABLE I: The Computation Cost of Our PPDCP-ABE

Scheme	Authorities Setup	Encryption	KeyGen	Decryption
PPDCP-ABE	$N(TE_{\mathbb{G}_\tau}+4TE_{\mathbb{G}})+(\sum_{i=1}^{N}3q_i)TE_{\mathbb{G}}$	$\lvert\mathcal{I}\rvert TE_{\mathbb{G}_\tau}+3\lvert\mathcal{I}\rvert TE_{\mathbb{G}}+(3\sum_{j\in\mathcal{I}}\ell_j)TE_{\mathbb{G}}$	$(9N+\lvert\tilde{U}\rvert)TE_{\mathbb{G}}$	$(4\lvert\mathcal{I}\rvert+\sum_{j\in\mathcal{I}}2\ell_j)TP+(\lvert\mathcal{I}\rvert+\sum_{j\in\mathcal{I}}\ell_j)TE_{\mathbb{G}}$

TABLE II: The Communication Cost of Our PPDCP-ABE

Scheme	Global Setup	Authorities Setup	Encryption	KeyGen
PPDCP-ABE	$3E_{\mathbb{G}}$	$(4N+\sum_{i}^{N}2q_i)E_{\mathbb{G}}+NE_{\mathbb{G}_\tau}$	$E_{\mathbb{G}_\tau}+(3\lvert\mathcal{I}\rvert+2\sum_{i\in\mathcal{I}}\ell_j)E_{\mathbb{G}}$	$(6N+\lvert\tilde{U}\rvert)E_{\mathbb{G}}$

$U(PP, PK_i, \mu, a_x \in \tilde{U}\bigcap\tilde{A}_i)$ $\qquad$ $\breve{A}_i(PP, PK_i, SK_i)$

1. Selects $k_1, k_2, d_1, d_2 \xleftarrow{\$} \mathbb{Z}_p$ and sets $d_u = d_1 d_2$.
Computes $\Theta_1 = A_i^{d_1}$, $\Theta_2 = g^{d_u}$, $\Theta_3 = h^{k_1}\mathfrak{g}^{\mu}$, $\Theta_4 = \Theta_3^{k_2}$,
$\Theta_5 = B_i^{k_2}$, $\Theta_6 = \mathfrak{g}^{\frac{1}{k_2}}$, $(\Psi_x^1 = T_x^{d_u}, \Psi_x^2 = Z_x^{d_u})_{a_x\in\tilde{U}\bigcap\tilde{A}_i}$
and $\Sigma_U = \mathsf{PoK}\{(k_1, k_2, d_1, d_u, \mu, (a_x \in \tilde{U}\bigcap\tilde{A}_i)):$
$\Theta_1 = A_i^{d_1}\wedge\ \Theta_2 = g^{d_u}\wedge\Theta_3 = h^{k_1}\mathfrak{g}^{\mu}\ \wedge\ \Theta_4 = \Theta_3^{k_2}\wedge$
$\Theta_5 = B_i^{k_2}\wedge\ e(\Theta_5,\Theta_6) = e(B_i,\mathfrak{g})\wedge\ (\wedge\frac{e(\Gamma_i^1,\Psi_x^1)}{e(\Gamma_i^2,\Psi_x^2)} =$
$e(g,\Psi_x^1)^{-a_x}\cdot\ \wedge e(h,\Psi_x^2)^{a_x}\cdot e(g,g)^{d_u})_{a_x\in\tilde{U}\bigcap\tilde{A}_i}\}$

$$\xrightarrow[\Theta_5,\Psi_x^1,\Psi_x^2,\Sigma_U]{\Theta_1,\Theta_2,\Theta_3,\Theta_4}$$

2. Selects $c_u, e_u \xleftarrow{\$} \mathbb{Z}_p$ and computes
$\Upsilon_1 = g^{c_u}, \Upsilon_2 = g^{\frac{1}{c_u}}, \Upsilon_3 = h^{c_u}$,
$\Upsilon_4 = h^{\frac{1}{c_u}}, \Upsilon_5 = g^{e_u}$,
$K_i' = g^{\alpha_i}\Theta_1^{e_u}\Theta_6^{c_u}(\Theta_4\Theta_5)^{\frac{1}{c_u}}$,
$(\Phi_x = (\Psi_x^2)^{e_u})_{a_x\in\tilde{U}\bigcap\tilde{A}_i}$ and
Σ_{A_i} $=\mathsf{PoK}\{(\alpha_i, c_u, e_u):$
$e(\Upsilon_1,\Upsilon_2) = e(g,g)\wedge\ \Upsilon_1 = g^{c_u}\wedge$
$\Upsilon_2 = g^{\frac{1}{c_u}}\ \wedge\Upsilon_3 = h^{c_u}\wedge\ \Upsilon_4 = h^{\frac{1}{c_u}}$
$e(\Upsilon_3,\Upsilon_4) = e(h,h)\wedge\ \Upsilon_5 = g^{e_u}\wedge$
$K_i' = g^{\alpha_i}\Theta_1^{e_u}\Theta_6^{c_u}(\Theta_4\Theta_5)^{\frac{1}{c_u}}$
$\wedge(\wedge(\Phi_x = (\Psi_x^2)^{e_u})_{a_x\in\tilde{U}\bigcap\tilde{A}_i}\}$.

$$\xleftarrow[\Upsilon 5,K_i',\Phi_x,\Sigma_{A_i}]{\Upsilon_1,\Upsilon_2,\Upsilon_3,\Upsilon_4}$$

3. Computes $K_i = \frac{K_i'}{\Upsilon_4^{k_1 k2}}$, $P_i = \Upsilon_5^{d_1}$, $L_i = \Upsilon_1^{\frac{1}{k_2}}$,
$R_i = \Upsilon_2^{k_2}$, $R_i' = \Upsilon_4^{k_2}$ and $\left(F_x = \Phi_x^{\frac{1}{d_2}}\right)_{a_x\in\tilde{U}\bigcap\tilde{A}_i}$

Fig. 2: PPKeyGen: Privacy-Preserving Key Generation Protocol

Then, U sends $(\tilde{k}_1, \tilde{k}_2, \tilde{k}_3, \tilde{k}_4, \tilde{k}_5, \tilde{k}_6)$ to $\breve{A}_i$.

4) $\breve{A}_i$ checks $e(\Theta_5,\Theta_6) = e(\Theta_5',\Theta_6') \stackrel{?}{=} e(B_i,\mathfrak{g})$, $\Theta_1' \stackrel{?}{=} A_i^{\tilde{d}_1}\Theta_1^{\eta}$, $\Theta_2' \stackrel{?}{=} g^{\tilde{d}_u}\Theta_2^{\eta}$, $\Theta_3' \stackrel{?}{=} h^{\tilde{k}_1}\mathfrak{g}^{\tilde{k}_3}\Theta_3^{\eta}$, $\Theta_4' \stackrel{?}{=} \Theta_3^{\tilde{k}_2}\Theta_4^{\eta}$, $\Theta_5' \stackrel{?}{=} B_i^{\tilde{k}_2}\Theta_5^{\eta}$, $\Theta_6' = \mathfrak{g}^{\tilde{k}_6}\Theta_6^{\eta}$, $(\Psi_x^4 \stackrel{?}{=} \mathfrak{h}^{\tilde{k}_4}g^{\tilde{k}_5}(\Psi_x^3)^{\eta}$, $\Psi_x^5 \stackrel{?}{=} (\frac{e(\Gamma_i^1,\Psi_x^1)}{e(\Gamma_i^2,\Psi_x^2)})^{\eta}\cdot e(g,\Psi_x^1)^{-\tilde{k}_5}\cdot e(h,\Psi_x^2)^{\tilde{k}_5}\cdot e(g,g)^{\tilde{d}_u})_{a_x\in\tilde{U}\bigcap\tilde{A}_i}$

If all the above equations hold, $\breve{A}_i$ selects $c_u, e_u, c_u', e_u', c_u'', l_u \xleftarrow{\$} \mathbb{Z}_p$ and computes $\Upsilon_1 = g^{c_u}$, $\Upsilon_2 = g^{\frac{1}{c_u}}$, $\Upsilon_3 = h^{c_u}$, $\Upsilon_4 = h^{\frac{1}{c_u}}$, $\Upsilon_5 = g^{e_u}$, $K_i' = g^{\alpha_i}\Theta_1^{e_u}\Theta_6^{c_u}(\Theta_4\Theta_5)^{\frac{1}{c_u}}$, $(\Phi_x = (\Psi_x^2)^{e_u})_{a_x\in\tilde{U}\bigcap\tilde{A}_i}$, $\Upsilon_1' = g^{c_u'}$, $\Upsilon_2' = g^{c_u''}$, $\Upsilon_3' = h^{c_u'}$, $\Upsilon_4' = h^{c_u''}$, $\Upsilon_5' = g^{e_u'}$, $K_i'' = g^{l_u}\Theta_1^{e_u'}\Theta_6^{c_u'}(\Theta_4\Theta_5)^{c_u''}$, $(\Phi_x' = (\Psi_x^2)^{e_u'})_{a_x\in\tilde{U}\bigcap\tilde{A}_i}$. Otherwise, $\breve{A}_i$ aborts.

$\breve{A}_i$ sends $(\Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, \Upsilon_1', \Upsilon_2', \Upsilon_3', \Upsilon_4', \Upsilon_5', K_i', K_i'', (\Phi_x, \Phi_x')_{a_x\in\tilde{U}\bigcap\tilde{A}_i})$ to U.

5) U selects $\tilde{\eta} \xleftarrow{\$} \mathbb{Z}_p$, and sends $\tilde{\eta}$ to $\breve{A}_i$.

6) $\breve{A}$ computes $\tilde{c}_u = c_u' - \tilde{\eta}c_u$, $\hat{c}_u = c_u'' - \frac{\tilde{\eta}}{c_u}$, $\tilde{e}_u = e_u' - \tilde{\eta}e_u$, and $\tilde{l}_u = l_u - \tilde{\eta}\alpha_i$. $\breve{A}_i$ sends $(\tilde{c}_u, \hat{c}_u, \tilde{e}_u, \tilde{l}_u)$ to U.

7) U checks $\Upsilon_1 \stackrel{?}{\neq} g$, $\Upsilon_2 \stackrel{?}{\neq} g$, $\Upsilon_3 \stackrel{?}{\neq} h$, $\Upsilon_4 \stackrel{?}{\neq} h$, $e(\Upsilon_1,\Upsilon_2) \stackrel{?}{=} e(g,g)$, $e(\Upsilon_3,\Upsilon_4) \stackrel{?}{=} e(h,h)$, $\Upsilon_1' \stackrel{?}{=} g^{\tilde{c}_u}\Upsilon_1^{\tilde{\eta}}$, $\Upsilon_2' \stackrel{?}{=} g^{\hat{c}_u}\Upsilon_2^{\tilde{\eta}}$, $\Upsilon_3' \stackrel{?}{=} \mathfrak{h}^{\tilde{c}_u}\Upsilon_3^{\tilde{\eta}}$, $\Upsilon_4' \stackrel{?}{=} \mathfrak{h}^{\hat{c}_u}\Upsilon_4^{\tilde{\eta}}$, $\Upsilon_5 \stackrel{?}{=} g^{\tilde{e}_u}\Upsilon_5^{\tilde{\eta}}$ and $K'' \stackrel{?}{=} g^{\tilde{l}_u}\Theta_1^{\tilde{e}_u}\Theta_6^{\tilde{c}_u}(\Theta_4\Theta_5)^{\hat{c}_u}K_i'^{\tilde{\eta}}$.
If all the above equations hold, U computes $K_i = \frac{K_i'\Upsilon_5^{\frac{1}{k_2}}}{\Upsilon_4^{k_1 k2}}$, $P_i = \Upsilon_6^{d_1}$, $L_i = \Upsilon_1^{\frac{1}{k_2}}$, $R_i = \Upsilon_2^{k_2}$, $R_i' = \Upsilon_4^{k_2}$ and $\left(F_x = \Phi_x^{\frac{1}{d_2}}\right)_{a_x\in\tilde{U}\bigcap\tilde{A}_i}$. Otherwise, U aborts.

F. Security of the Proposed PPKeyGen Protocol

Theorem 3. *The privacy-preserving key extract protocol* PPKeyGen *in Fig. 2 is both leak-free and selective-failure blind under the q-SDH assumption.*

Proof: We first prove that the PPKeyGen protocol is leak-free, then prove that it is selective-failure blind.

Leak-Freeness. It requires that there exist an efficient simulator $\bar{U}$ such that no efficient distinguisher $\mathcal{D}$ can distinguish the real world experiment (where the malicious user $\mathcal{U}$ is executing the PPKeyGen algorithm with the honest authority $\breve{A}_i$) from the ideal world experiment (where $\breve{A}_i$ is executing the algorithm KeyGen with a trusted party). $\bar{U}$ simulates the communication between $\mathcal{U}$ and $\breve{A}_i$ by passing the input of $\mathcal{D}$ to $\mathcal{U}$ and the output of $\mathcal{U}$ to $\mathcal{D}$. The real world experiment is as follows.

1) $\bar{U}$ sends the public parameters $params$ and the public key PK_i of $\breve{A}_i$ to $\mathcal{U}$.

IEEE Transactions on Information Forensics and Security,(Volume:10 , Issue: 3),March 2015.

2) $\mathcal{U}$ must output $(\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5, (\Psi_x^1, \Psi_x^2)_{a_x \in \tilde{\mathcal{U}} \bigcap \tilde{A}_i})$, and prove $\mathsf{PoK}\{(k_1, k_2, d_1, d_u, \mu, (a_x \in \tilde{\mathcal{U}} \bigcap \tilde{A}_i)) : \Theta_1 = A_i^{d_1} \wedge \Theta_2 = g^{d_u} \wedge \Theta_3 = \Gamma_3 = \mathfrak{h}^{k_1}\mathfrak{g}^{\mu}, \wedge \Theta_4 = \Theta_3^{k_2} \wedge \Theta_5 = g^{k_2} \wedge (\wedge \frac{e(\Gamma_i^1, \Psi_x^1)}{e(\Gamma_i^2, \Psi_x^2)} = e(g, \Psi_x^1)^{-a_x} \cdot e(h, \Psi_x^2)^{a_x} \cdot e(g, g)^{d_u})_{a_x \in \tilde{\mathcal{U}} \bigcap \tilde{A}_i}\}$. If the proof fails, $\bar{U}$ aborts; otherwise, $\bar{U}$ can obtains $(d_1, d_u, k_1, k_2, \mu, (a_x \in \tilde{\mathcal{U}} \bigcap \tilde{A}_i))$ by using the rewind technique.
3) $\bar{U}$ can computes $Z_x = (\Psi_x^2)^{\frac{1}{d_u}}$ for $a_x \in \tilde{\mathcal{U}} \bigcap \tilde{A}_i$, and sends $\left(\mu, (Z_x)_{a_x \in \tilde{\mathcal{U}} \bigcap \tilde{A}_i}\right)$ to the trusted party. The latter runs the KeyGen algorithm to generate secrete key $SK = \left(K_i, P_i, L_i, L_i, R_i, R_i, (F_x)_{a_x \in \tilde{\mathcal{U}} \bigcap \tilde{A}_i}\right)$.
4) $\bar{U}$ computes $\Upsilon_1 = L_i^{k_2}$, $\Upsilon_2 = R_i^{\frac{1}{k_2}}$, $\Upsilon_3 = L_i'^{k_2}$, $\Upsilon_4 = R_i'^{\frac{1}{k_2}}$, $\Upsilon_5 = P_i^{\frac{1}{d_1}}$, $K_i' = K_i(\Upsilon_4)^{k_1 k_2}$ and $\Phi_x = F_x^{\frac{d_u}{d_1}}$.

If $\left(K_i, P_i, L_i, L_i, R_i, R_i, (F_x)_{a_x \in \tilde{\mathcal{U}} \bigcap \tilde{A}_i}\right)$ is a correct secret key from the trusted party in the ideal world experiment, $\left(\Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, K_i', (\Phi_x)_{a_x \in \tilde{\mathcal{U}} \bigcap \tilde{A}_i}\right)$ is correct secret key from $\breve{A}_i$ in the real world experiment. Hence, $(K_i, P_i, L_i, L_i, R_i, R_i, (F_x)_{a_x \in \tilde{\mathcal{U}} \bigcap \tilde{A}_i})$ and $(\Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, K_i', (\Phi_x)_{a_x \in \tilde{\mathcal{U}} \bigcap \tilde{A}_i})$ are identically distributed. Therefore, no efficient distinguisher $\mathcal{D}$ can distinguish the real world experiment from the ideal world experiment.

Selective-Failure Blindness. The malicious authority $\mathcal{A}_i$ submits the public key PK_i and two pairs of GIDs and attributes: $(\mu_0, \tilde{U}_0)$ and $(\mu_1, \tilde{U}_1)$. Then, a bit $\vartheta \in \{0, 1\}$ is selected. $\mathcal{A}_i$ can black-box access the orales

$$U\left(params, \mu_0, \tilde{U}_0, PK_i, decom_i, (decom_{i,j})_{a_{i,j} \in \tilde{U}_0 \bigcap \tilde{\mathcal{A}}_i}\right)$$

and

$$U\left(params, \mu_1, \tilde{U}_1, PK_i, decom_i, (decom_{i,j})_{a_{i,j} \in \tilde{U}_1 \bigcap \tilde{\mathcal{A}}_i}\right).$$

After this, U executes the $\mathsf{PPKeyGen}$ algorithm with $\mathcal{A}_i$ where $\mathcal{A}_i$ plays the role of the authority $\breve{A}_i$. U outputs secret keys SK_{U_0} and SK_{U_1} for $(\mu_0, \tilde{U}_0)$ and $(\mu_1, \tilde{U}_1)$, respectively. If $SK_{U_0} \neq \perp$ and $SK_{U_1} \neq \perp$, $\mathcal{A}_i$ is given (SK_{U_0}, SK_{U_1}); if $SK_{U_0} = \perp$ and $SK_{U_1} \neq \perp$, $\mathcal{A}_i$ is given $(\epsilon, \perp)$; if $SK_{U_0} \neq \perp$ and $SK_{U_1} = \perp$, $\mathcal{A}_i$ is given $(\perp, \epsilon)$; if $SK_{U_0} = \perp$ and $SK_{U_1} = \perp$, $\mathcal{A}_i$ is given (ϵ, ϵ). Finally, $\mathcal{A}_i$ will output his guess ϑ' on ϑ.

In the $\mathsf{PPKeyGen}$ protocol, U sends $(\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5, (\Psi_x^1, \Psi_x^2)_{a_x \in \tilde{U}_b \bigcap \tilde{A}_i})$, and proves $\mathsf{PoK}\{(k_1, k_2, d_u, \mu, (a_x \in \tilde{U} \bigcap \tilde{A}_i)) : \Theta_1 = A_i^{d_u} \wedge \Theta_2 = g^{d_u} \wedge \Theta_3 = h^{k_1}\mathfrak{g}^{\mu}, \wedge \Theta_4 = \Theta_3^{k_2} \wedge \Theta_5 = B_i^{k_2} \wedge e(\Theta_5, \Theta_6) = e(B_i, \mathfrak{g}) \wedge (\wedge \frac{e(\Gamma_i^1, \Psi_x^1)}{e(\Gamma_i^2, \Psi_x^2)} = e(g, \Psi_x^1)^{-a_x} \cdot e(h, \Psi_x^2)^{a_x} \cdot e(g, g)^{d_u})_{a_x \in \tilde{U}_b \bigcap \tilde{A}_i}\}$. Up to this point, $\mathcal{A}_i$ runs one or both the oracles. So far, $\mathcal{A}_i$' view on the two oracles are computationally undistinguishable; otherwise, the hiding property of the commitment scheme and the zero-knowledge property of the zero-knowledge proof are broken. If $\mathcal{A}_i$ can use any computing strategy to output the secret key $(\Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, K_i', (\Phi_x)_{a_x \in \tilde{U}_b \in \tilde{A}_i})$ for the first oracle, we show that $\mathcal{A}_i$ can predict SK_{U_b} without the interactions with the two oracles.

TABLE III: The Computation Cost of the PPKeyGen Algorithm

Algorithm	User U	Authority $\breve{A}_i$
PP-KeyGen	$(4 + 3\lvert\tilde{U} \bigcap \tilde{A}_i\rvert)TP+$ $(35 + 7\lvert\mathcal{I}\rvert)TE_{\mathbb{G}}+$ $3\lvert\tilde{U} \bigcap \tilde{A}_i\rvert E_{\mathbb{G}_\tau}$	$(3 + 5\lvert\tilde{U} \bigcap \tilde{A}_i\rvert)TP+$ $(18 + 5\lvert\tilde{U} \bigcap \tilde{A}_i\rvert)TE_{\mathbb{G}}+$ $4\lvert\tilde{U} \bigcap A_i\rvert TE_{\mathbb{G}_\tau}$

1) $\mathcal{A}_i$ checks $\mathsf{PoK}\{(\alpha_i, c_u, e_u) : \Upsilon_1 = g^{c_u} \wedge \Upsilon_2 = g^{\frac{1}{c_u}} \wedge e(\Upsilon_1, \Upsilon_2) = e(g, g) \wedge \Upsilon_3 = h^{c_u} \wedge \Upsilon_4 = h^{\frac{1}{c_u}} \wedge e(\Upsilon_3, \Upsilon_4) = e(h, h) \wedge K_i' = \Upsilon_5 = g^{e_u} \wedge K_i' = g^{\alpha_i}\Theta_1^{e_u}\Theta_6^{c_u}(\Theta_4\Theta_5)^{\frac{1}{c_u}} \wedge (\wedge(\Phi_x = (\Psi_x^2)^{e_u})_{a_x \in \tilde{U} \bigcap \tilde{A}_i}\}$. If the proof fails, $\mathcal{A}$ sets $SK_{U_0} = \perp$.
2) $\mathcal{A}_i$ generates a different $(\Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, K_i', (\Phi_x)_{a_x \in \tilde{U}_b \in \tilde{A}_i})$ for the second oracle and a zero-knowledge proof $\mathsf{PoK}\{(\alpha_i, c_u, e_u) : \Upsilon_1 = g^{c_u} \wedge \Upsilon_2 = g^{\frac{1}{c_u}} \wedge e(\Upsilon_1, \Upsilon_2) = e(g, g) \wedge \Upsilon_3 = h^{c_u} \wedge \Upsilon_4 = h^{\frac{1}{c_u}} \wedge e(\Upsilon_3, \Upsilon_4) = e(h, h) \wedge K_i' = \Upsilon_5 = g^{e_u} \wedge K_i' = g^{\alpha_i}\Theta_1^{e_u}\Theta_6^{c_u}(\Theta_4\Theta_5)^{\frac{1}{c_u}} \wedge (\wedge(\Phi_x = (\Psi_x^2)^{e_u})_{a_x \in \tilde{U} \bigcap \tilde{A}_i}\}$. If the proof fails, $\mathcal{A}_i$ sets $SK_{U_1} = \perp$.
3) If either test failed, then : if $SK_{U_0} = \perp$ and $SK_{U_1} \neq \perp$, outputs $(\epsilon, \perp)$. If $(SK_{U_0}) \neq \perp$ and $SK_{U_1} = \perp$, outputs $(\perp, \epsilon)$. If both tests failed, outputs $(\perp, \perp)$.
4) If both tests succeeded, $\mathcal{A}_i$ executes $\mathsf{PPKeyGen}$ with himself on inputs $(\mu_0, \tilde{U}_0)$ and $(\mu_1, \tilde{U}_1)$. If either protocol fails, $\mathcal{A}_i$ aborts. Otherwise, $\mathcal{A}_i$ outputs (SK_{U_1}, SK_{U_2}).

The prediction on $(\mu_0, \tilde{U}_0)$ and $(\mu_1, \tilde{U}_1)$ is correct, and has the identical distribution with the oracle. So, $\mathcal{A}_i$ can output the valid secret key which is the same as $\mathcal{U}$ obtains from $\mathsf{PPKeyGen}(U \leftrightarrow \breve{A}_i)$ when the both the proofs are correct as $\mathcal{A}_i$ performs the same work as U. Therefore, if $\mathcal{A}_i$ can predict the outputs of the two oracles, his advantage in distinguishing

$$U\left(params, \mu_0, \tilde{U}_0, PK_i, decom_i, (decom_{i,j})_{a_{i,j} \in \tilde{U}_0 \bigcap \tilde{\mathcal{A}}_i}\right)$$

from

$$U(params, \mu_1, \tilde{U}_1, PK_i, decom_i, (decom_{i,j})_{a_{i,j} \in \tilde{U}_1 \bigcap \tilde{\mathcal{A}}_i})$$

is the same without the final output. Hence, the advantage of $\mathcal{A}_i$ should come from the received $(\Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, K_i', (\Phi_x)_{a_x \in \tilde{U}_b \in \tilde{A}_i})$ and the proof $\mathsf{PoK}\{(\alpha_i, c_u, e_u) : \Upsilon_1 = g^{c_u} \wedge \Upsilon_2 = g^{\frac{1}{c_u}} \wedge e(\Upsilon_1, \Upsilon_2) = e(g, g) \wedge \Upsilon_3 = h^{c_u} \wedge \Upsilon_4 = h^{\frac{1}{c_u}} \wedge e(\Upsilon_3, \Upsilon_4) = e(h, h) \wedge K_i' = \Upsilon_5 = g^{e_u} \wedge K_i' = g^{\alpha_i}\Theta_1^{e_u}\Theta_6^{c_u}(\Theta_4\Theta_5)^{\frac{1}{c_u}} \wedge (\wedge(\Phi_x = (\Psi_x^2)^{e_u})_{a_x \in \tilde{U} \bigcap \tilde{A}_i}\}$. By the hiding property of the commitment and the witness undistinguishable property, $\mathcal{A}_i$ cannot distinguish one from the other with non-negligible advantage. ∎

G. Efficiency of The Proposed KeyGen Protocol

We describe the computation cost and communication of the $\mathsf{PPKeyGen}$ algorithm in Table III and Table IV, respectively. $\tilde{U}$ and $\tilde{A}_i$ are denoted as the set of attributes held by U and the set of attributes monitored by the authority A_i, respectively.

TABLE IV: The Communication Cost of The PPKeyGen Algorithm

Algorithm	$U \rightarrow \check{A}_i$	$U \leftarrow \check{A}_i$
PP-KeyGen	$9E_p + (12 + 2\vert\tilde{U} \bigcap \tilde{A}_i\vert)E_{\mathbb{G}} + \vert\tilde{U} + \tilde{A}_i\vert E_{\mathbb{G}_\tau}$	$5E_p + (12 + 2\vert\tilde{U} \bigcap \tilde{A}_i\vert E_{\mathbb{G}}$

H. Security of the Proposed PPDCP-ABE

By **Theorem** 2 and **Theorem** 3, we have the following theorem.

Theorem 4. *Our privacy-preserving decentralized ciphertext-policy attribute-based encryption (PPDCP-ABE) scheme* $\tilde{\prod} = (\mathsf{Global\ Setup}, \mathsf{Authority\ Setup}, \mathsf{Encrypt}, \mathsf{PPKeyGen}, \mathsf{Decrypt})$ *is secure in the selective-access structure model under the decisional q-PBDHE assumption and q-SDH assumption.*

V. Conclusion

Some PPMA-ABE schemes have been proposed to protect users' privacy and reduce the trust on the central authority. Nevertheless, only the privacy of the GID was considered in the existing scheme. Since sensitive attributes can also reveal the users' identities, existing schemes cannot provide a full solution to protect users' privacy in MA-ABE schemes. In this paper, we proposed a PPDCP-ABE scheme where both the privacy of the GID and the attributes are concerned. In our scheme, a central authority is not required and multiple authorities can work independently without any cooperation. A user can convince the authorities that the attributes for which he is obtaining secret keys are monitored by them without showing the attributes to them. Therefore, our scheme provides a perfect solution for the privacy issues in MA-ABE schemes.

As for future research direction regarding PPDCP-ABE, it would be interesting to construct a fully secure PPDCP-ABE scheme since the scheme proposed in this paper is selectively secure.

Acknowledgement

We would like to thank the anonymous reviewers for useful comments. The first author was partially supported by National Natural Science Foundation of China (Grant No. 61300213), National Center for International Joint Research on E-Business Information Processing (Grant No. 2013B01035) and Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, China. The second author was partially supported by Australia Research Council Discovery Project (DP130101383).

References

[1] J. Han, W. Susilo, Y. Mu, J. Zhou, and M. H. Au, "PPDCP-ABE: Privacy-preserving decentralized cipher-policy attribute-based encryption," in *Proc. ESORICS'14*, vol. 8713 of *Lecture Notes in Computer Science*, pp. 73–90, Springer, Switzerland, 2014.

[2] P. Bichsel, J. Camenisch, T. Groβ, and V. Shoup, "Anonymous credentials on a standard java card," in *Proc. CCS'09*, pp. 600–610, ACM, 2009.

[3] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. EUROCRYPT'05*, vol. 3494 of *Lecture Notes in Computer Science*, pp. 457–473, Springer, Heidelberg, 2005.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. S& P'07*, pp. 321–334, IEEE, 2007.

[5] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. CCS'07*, pp. 456–465, ACM, 2007.

[6] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. CCS'06*, pp. 89 – 98, ACM, 2006.

[7] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. CCS'07*, pp. 195–203, ACM, 2007.

[8] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proc. EUROCRYPT'10*, vol. 6110 of *Lecture Notes in Computer Science*, pp. 62–91, Springer, Heidelberg, 2010.

[9] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. PKC'11*, vol. 6571 of *Lecture Notes in Computer Science*, pp. 53–70, Springer, Heidelberg, 2011.

[10] M. Chase, "Multi-authority attribute based encryption," in *Proc. TCC'07*, vol. 4392 of *Lecture Notes in Computer Science*, pp. 515–534, Springer, Heidelberg, 2007.

[11] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. EUROCRYPT'11*, vol. 6632 of *Lecture Notes in Computer Science*, pp. 568–588, Springer, Heidelberg, 2011.

[12] M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. CCS'09*, pp. 121–130, ACM, 2009.

[13] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2150–2162, 2012.

[14] H. Qian, J. Li, and Y. Zhang, "Privacy-preserving decentralized ciphertext-policy attribute-based encryption with fully hidden access structure," in *Proc. ICICS'13*, vol. 8233 of *Lecture Notes in Computer Science*, pp. 363–372, Springer, Heidelberg, 2013.

[15] J. Li, Q. Huang, X. Chen, S. S. M. Chow, D. S. Wong, and D. Xie, "Multi-authority ciphertext-policy attribute-based encryption with accountability," in *Proc. ASIACCS'11*, pp. 386–390, ACM, 2011.

[16] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *Proc. PKC'10*, vol. 6056 of *Lecture Notes in Computer Science*, pp. 19–34, Springer, Heidelberg, 2010.

[17] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," in *Proc. INDOCRYPT'08*, vol. 5365 of *Lecture Notes in Computer Science*, pp. 426–436, Springer, Heidelberg, 2008.

[18] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," in *Proc. EUROCRYPT'99*, vol. 1592 of *Lecture Notes in Computer Science*, pp. 295–310, Springer, Heidelberg, 1999.

[19] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust threshold dss signatures," in *Proc. EUROCRYPT'96*, vol. 1070 of *Lecture Notes in Computer Science*, pp. 354–371, Springer, Heidelberg, 1996.

[20] S. Müller, S. Katzenbeisser, and C. Eckert, "Distributed attribute-based encryption," in *Proc. ICISC'08*, vol. 5461 of *Lecture Notes in Computer Science*, pp. 20–36, Springer, Heidelberg, 2008.

[21] Z. Liu, Z. Cao, Q. Huang, D. S. Wong, and T. H. Yuen, "Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles," in *Proc. ESORICS'11*, vol. 6879 of *Lecture Notes in Computer Science*, pp. 278–297, Springer, Heidelberg, 2011.

[22] M. Naor, B. Pinkas, and O. Reingold, "Distributed pseudo-random functions and KDCs," in *Proc. EUROCRYPT'99*, vol. 1592 of *Lecture Notes in Computer Science*, pp. 327–346, Springer, Heidelberg, 1999.

[23] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *Proc. EUROCRYPT'01*, vol. 2045 of *Lecture Notes in Computer Science*, pp. 93–118, Springer, Heidelberg, 2001.

[24] G. Persiano and I. Visconti, "An efficient and usable multi-show non-transferable anonymous credential system," in *Proc. FC'04*, vol. 3110 of *Lecture Notes in Computer Science*, pp. 196–211, Springer, Heidelberg, 2004.

[25] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Proc. EUROCRYPT'04*, vol. 3027 of *Lecture Notes in Computer Science*, pp. 56–73, Springer, Heidelberg, 2004.

[26] A. Beime, *Secure Schemes for Secret Sharing and Key Distribution*. Phd thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

IEEE Transactions on Information Forensics and Security,(Volume:10 , Issue: 3),March 2015.

[27] T. P. Pedersen, “Non-interactive and information-theoretic secure verifiable secret sharing,” in *Proc. CRYPTO’91*, vol. 576 of *Lecture Notes in Computer Science*, pp. 129–140, Springer, Heidelberg, 1992.

[28] J. Camenisch and M. Stadler, “Efficient group signature schemes for large groups,” in *Proc. CRYPTO’97*, vol. 1294 of *Lecture Notes in Computer Science*, pp. 410–424, Springer, Heidelberg, 1997.

[29] J. Camenisch, R. Chaabouni, and A. Shelat, “Efficient protocols for set membership and range proofs,” in *Proc. ASIACRYPT’08*, vol. 5350 of *Lecture Notes in Computer Science*, pp. 234–252, Springer, Heidelberg, 2008.

[30] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy, “Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data,” in *Proc. PKC’09*, vol. 5443 of *Lecture Notes in Computer Science*, pp. 196–214, Springer, Heidelberg, 2009.

[31] M. Green and S. Hohenberger, “Blind identity-based encryption and simulatable oblivious transfer,” in *Proc. ASIACRYPT’07*, vol. 4833 of *Lecture Notes in Computer Science*, pp. 265–282, Springer, Heidelberg, 2007.

Jinguang Han received his Ph.D from University of Wollongong, Australia, in 2013. He currently is an associate professor in Jiangsu Provincial Key Laboratory of EBusiness, Nanjing University of Finance and Economics, China. His main research interests include cryptography and information security. He has served as a program committee member in over 10 international conferences. He has published more than 20 research papers in refereed international journals and conferences. He is a member of the IEEE.

Willy Susilo received a Ph.D. in Computer Science from University of Wollongong, Australia. He is a Professor at the School of Computer Science and Software Engineering and the co-director of Centre for Computer and Information Security Research (CCISR) at the University of Wollongong. He previously held the prestigious ARC Future Fellow awarded by the Australian Research Council (ARC). His main research interests include cryptography and information security. His main contribution is in the area of digital signature schemes. He has served as a program committee member in dozens of international conferences. He has published numerous publications in the area of digital signature schemes and encryption schemes. He is a senior member of IEEE since 2001.

Yi Mu received his PhD from the Australian National University in 1994. He currently is a professor, Head of School of Computer Science and Software Engineering and the co-director of Centre for Computer and Information Security Research at University of Wollongong, Australia. His current research interests include network security, computer security, and cryptography. Yi Mu is the editor-in-chief of International Journal of Applied Cryptography and serves as associate editor for nine other international journals. He is a senior member of the IEEE and a member of the IACR. He is a senior member of IEEE since 2000.

Jianying Zhou received the PhD degree in information security from the University of London in 1997. He is currently a senior scientist at the Institute for Infocomm Research, and heads the Infocomm Security Department. His research interests include applied cryptography, computer and network security, mobile and wireless communications security. He is a cofounder and steering committee member of International Conference on Applied Cryptography and Network Security.

Man Ho Au is an assistant professor in the Department of Computing at the Hong Kong Polytechnic University, Hong Kong. Before moving to Hong Kong, he has been a lecturer at the School of Computer Science and Software Engineering, University of Wollongong, Australia. He obtained his bachelor (2003) and master (2005) degrees in the Department of Information Engineering, the Chinese University of Hong Kong (CUHK). He received his PhD from the University of Wollongong (UOW) in 2009.

He works in the area of information security. In particular, his research interests include applying public-key cryptographic techniques to systems with security and privacy concerns. He has published over 70 referred journal and conference papers, including 2 papers in the ACM CCS conference that were named as ’Runners-up for PET Award 2009: Outstanding Research in Privacy Enhancing Technologies.