IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 1, 2017, accepted November 26, 2017, date of publication December 6, 2017,
date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2780763

PriGuarder: A Privacy-Aware Access Control Approach Based on Attribute Fuzzy Grouping in Cloud Environments

LI LIN[1,2,3], (Member, IEEE), TING-TING LIU[1,2], SHUANG LI[1,2], CHATHURA M. SARATHCHANDRA MAGURAWALAGE[4], AND SHAN-SHAN TU[1,2]

[1]College of Computer Science, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
[2]Beijing Key Laboratory of Trusted Computing, Beijing 100124, China
[3]National Engineering Laboratory for Classified Information Security Protection, Beijing 100124, China
[4]Department of Computer Science and Electronic Engineering, University of Essex, Essex CO4 3SQ, U.K.

Corresponding author: Li Lin (e-mail: linli_2009@bjut.edu.cn)

This work was supported in part by the National Science Foundation of China under Grant 61502017, in part by the Scientific Research Common Program of Beijing Municipal Commission of Education under Grant KM201710005024, and in part by the National Key Research and Development Program of China under Grant 2016YFB0800204.

ABSTRACT Data privacy protection is crucial to cloud computing since privacy leakage may prevent users from using cloud services. To ensure data privacy, we propose *PriGuarder*, a novel privacy-aware access control method. This method spans the three stages of a cloud service, i.e., user registration, data creation, and data access. At each stage, users can choose two modes to interact with the cloud service provider, i.e., direct or indirect. With the indirect mode, an attribute fuzzy grouping scheme is introduced to ensure user identity privacy and attribute privacy in all the three stages. Furthermore, exploiting data encryption and timestamp techniques, new access control protocols are proposed to regulate interactions between users and the cloud service provider. We illustrate the use of our method in the context of Amazon S3. Theoretical analysis and comprehensive simulation experiments have been conducted, which demonstrate the efficacy of *PriGuarder*.

INDEX TERMS Data privacy protection, access control, attribute fuzzy grouping.

I. INTRODUCTION

In cloud computing, user data may contain user privacy, such as enterprise customer information, hospital patient information, and users dating [1]–[3], and hence should be well protected. In addition, a malicious cloud service provider (CSP) or an attacker could intercept users' operations and then access user identity information or tamper user data, which may result in serious consequences [4]. Therefore, it is crucial to ensure user privacy in cloud computing. Access control has been considered as an effective solution to ensure user privacy [5].

However, designing access control is particularly challenging in the cloud computing setting. First, most of the traditional approaches ensure user identity privacy by using attribute-based techniques, which fail to protect user attribute privacy. Second, it still has the risk of identity disclosure by correlation analysis of user attribute information. Third, the access control methods commonly used can't ensure the reliability of anonymous users. An attacker can still obtain data in the cloud by pretending to be a legitimate user.

There have been a lot of research efforts. In [1], [6]–[8], attribute-based encryption (ABE) methods were proposed to protect cloud data privacy. By associating keys for encryption and decryption with users' attribute set, ABE can protect data privacy in the cloud and improve the flexibility of access control. But it has required users to provide their attribute certificates, and failed to protect user attribute privacy. Aimed at the problem, [9]–[13] have put forward some anonymous access control (AAC) methods. However, the existing AAC methods are achieved by group authentication methods and fails to realize on-demand fee [12]. [14]–[17] have proposed some privacy protection methods by combining the advantages of ABE and AAC, but these works still ignore the privacy disclosure on users' attribute. For example, suppose

that a FBI member uses cloud storage service and accesses his data in the cloud by providing his FBI credentials, an attacker can learn the users' FBI attribute and pose a threat to his data when the attacker can monitor the user's operations on access to cloud. In addition, there have been a lot of people who upload their personal diaries or photos to cloud storage servers, which are offered by Baidu, Yahoo and other Internet companies. In order to complete registration, people are often required to provide some information such as graduate school, work place and so on. Although a cloud user does not provide identity information, an attacker may obtain the user's exact identity by data mining and correlation analysis techniques of his attribute information.

In response to the limitations with existing approaches, we propose a novel Privacy-aware Access Control method based on Attribute Fuzzy Grouping, called *PriGuarder*. This method spans the three stages of a cloud service, i.e., *user registration*, *data creation* and *data access*. At each stage, it provides user two kinds of interaction — direct mode or indirect mode. With the indirect mode, an Attribute Fuzzy Grouping (AFG) scheme is introduced to ensure user identity privacy and attribute privacy. Furthermore, exploiting data encryption and timestamp techniques, new access control protocols are proposed to regulate interaction between users and the cloud service providers. We illustrate the use of our method in the context of Amazon S3. Our principal contributions are summarized below.

- Based on trusted third party (TTP) privacy protection framework, *PriGuarder* can not only support real-name access but also guarantee anonymous access control.
- *PriGuarder* leverages an AFG technique, which can realize the attribute-based access control and also support user attribute privacy protection. This issue has been ignored by most of existing AAC schemes.
- Rigid theoretical analysis as well as experimental evaluation have been presented to show the effectiveness of *PriGuarder* with respect to privacy protection.

The rest of the paper is organized as follows. Section II presents related work. In Section III, we present the problem description. Section IV gives the design details of the proposed method. Section V gives an example to illustrate the use of the proposed method. In Section VI, we present theoretical analysis on the proposed method. In Section VII, the experimental results are discussed. In Section VIII, we conclude the paper and present some future work.

II. RELATED WORK

Existing privacy protection approaches for cloud computing can be divided into three categories, i.e., ABE, AAC or hybrid.

Goyal *et al.* [18] proposed an ABE technology, where user's public key and private key are associated with user's attributes. In the aspect of ABE researches, there are two kinds of schemes: Key-policy ABE (KP-ABE) [18] and Ciphertext-policy ABE (CP-ABE) [19], [20]. In KP-ABE, data is stored in the cloud server with encryption. The user, who wants to access data, must match his attributes and keys with those sent by attribute authority. In CP-ABE, user identity is expressed as a set of user attributes and encrypted data is associated with access control structure. The user who wants to access data must follow an access protocol based on a kind of tree structure, which makes attributes as tree's leaves and have AND, OR and Other as the thresholds of monotone access structures. Whether user unlock the ciphertext or not depends on his attributes matched the corresponding access protocol structure. Zhang and Chen [6] proposed a HDFS-based attribute encryption method, which combines a proxy re-encryption with inert reinsurance confidential method and can reduce in the computational cost of the data owner. Huang et al. [7] proposed a multi-role user access control solution by employing attribute-based encryption technique. Nabeel and Bertino [8] proposed an incremental encryption technology to reduce the re-encrypting costs caused by the change of access control policy. Chase and Chow [22] proposed a multi-certified ABE method, where multiple KDCs are coordinated by a unified authority that assign attributes and keys to users. Lin et al. [23] proposed a multi-authority ABE protocol without the need for a credible authority. However, it requires multiple KDC with full knowledge of each user properties. Lewko and Waters [24] put forward a decentralized ABE method, which doesn't require a trusted server and users can have multiple attributes. Green et al. [25] proposed a decrypt task agent method to reduce the computation overhead. Compared with the distributed ways, this approach that combines a single agent with a centralized KDC still needs to verify users' identity and don't meet the demand of the user's anonymous access.

To support anonymous access, Yang et al. [9] adopted a centralized KDC access control method to support user authentication. Maji et al. [10] raised an attribute-based authentication and centralized KDC AAC method, and used distributed KDC to implement anonymous access in [11]. However, the work mentioned above ignore the issues caused by anonymous access. Under anonymous access, it is difficult for cloud service providers to confirm the misuse of resources.

Some other researchers have presented comprehensive technical solutions that make full use of the advantage of both attributes-based encryption and anonymous access. For example, Jensen et al. [12] presented a data anonymization method to prevent cloud service providers from abusing user data. Meanwhile, the work has achieved reliable AAC and accountability by adopting a ring and group signature technique. In view of the DaaS, Jia et al. [14] presented a method to protect user privacy on the premise without TTP. Yang and Jia [15] proposed a privacy protection scheme based on multi-authority CP-ABE, which can solve the problem of attribute revocation. Gao [3] and Fan [4] also presented ABE methods to protect user identity privacy. However, all the work discussed above take a set of user attribute information as his ID, there still exists a problem of attributes privacy leakage.

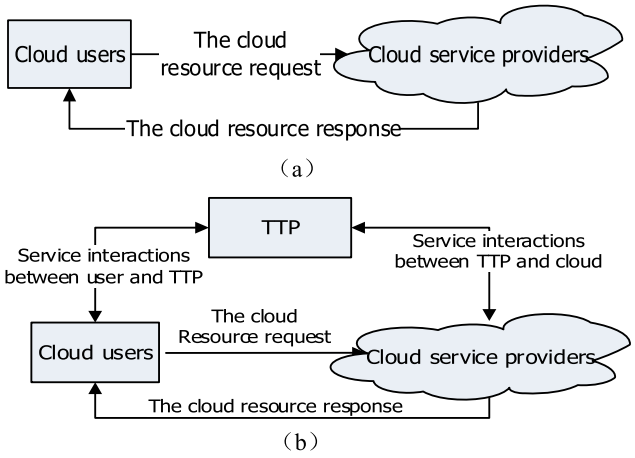

FIGURE 1. Typical cloud computing application scenarios. (a) Direct interaction. (b) Interaction based on the TTP.

III. PROBLEM DESCRIPTION AND PRELIMINARY

A. SYSTEM MODEL

In general, a cloud service system include three kinds of entities, cloud users, cloud service provider (CSP) and Trusted Third party (TTP). The cloud user is the party that utilizes cloud service, for example, retrieving the specific data, getting the computing results, and accessing the shared data. The CSP has significant storage space and computation capacities, and is responsible for providing all the aforementioned services and implementing access control to the cloud data. The TTP is responsible for regulating of CSP's performance or QoS. For instance, a TTP like TapInSystems, TechOut and Hyperic can detect the performance of amazon's S3, such as service availability, response time, throughput, etc. And a TTP also can be responsible for providing the third party payment service, such as Alipay [26] is used to realize the third-party payment by Baidu Cloud in China. Besides, there also exist some other TTPs that can offer certain privacy protection service. For example, Chrome cloud browser has provided a third-party cookies service, which can prevent cloud service websites recording users' personal information and pushing them to CSPs by cookies.

There are two modes,*direct interaction* or *interaction based on the TTP*, in current cloud computing applications. In the direct interaction, as shown in Fig. 1 (a), users directly request cloud resources from a CSP, and the CSP responses the request after it has completed user identity authentication. This mode is more suitable to small and medium enterprises that plan to build a private cloud environment, like VSPEX – Eblock offered by EMC. In the interaction based on the TTP, as shown in Fig. 1 (b), the cloud resource requests and responses between users and CSPs need help from a TTP.

In this paper, a TTP is introduced to do similar things and users are allowed to choose the above two interaction modes on demand. Either way, users can create data and declare others' access mode and permissions to data that he has created. Our main goal is to propose a privacy-aware access control method, where users' identity and attribute privacy can be protected under the support of TTP.

To apply our approach, the following reasonable assumptions must hold.

- CSPs are honest but may be curious, which may view user data information contents out of curiosity, but not modify users' data.
- CSPs and TTPs must not collude.
- Attribute-based access control techniques are used in cloud environments.
- Users' access rights to data in cloud include read, write, read-only and write-only.
- All data exchange between entities (i.e., CSPs and users, or TTPs and users) are protected by security protocols.

B. PRELIMINARY OF AFG-BASED ACCESS CONTROL POLICY

We have made formal description on attribute-based access control policy in our previous work [27], where S stands for access subject set,Astands for operation set,O stands for access object set. Taking $s \in S$ as an example, s is described by a tuple $(sa_1, sa_2, \ldots, sa_l)$, where $sa_i \in Dom(SA_i)(i = 1, \ldots, l,\ l \in N)$ and $Dom(SA_i)$ is denoted as the value domain of subjects' *ith* attribute. In this paper, we propose an attribute fuzzy grouping (AFG) scheme to group an access entity's attribute faintly and distribute each access entity a fuzzy identity. The basic idea is to increase the difficulty where the specific attribute information of cloud users can be deduced by malicious CSPs or other attackers. That means, if there are no attackers can find the exact attribute information, user attribute privacy is protected. We will introduce the corresponding scheme in Section IV. Here we give definitions of terms and symbols.

AFG-based access control policy is a kind of extension from attribute-based access control policy. Based on our previous work [27], we firstly define fuzzy grouping domain of subject's attribute as follows.

Definition 1: Let $SATD_i = \{Dom(SA_i)_j | j = 1, \ldots, m\}$, where $Dom(SA_i) = \cup_{j=1}^{m} Dom(SA_i)_j, Dom(SA_i)_p \cap Dom(SA_i)_q = \emptyset, p, q \in N, p \neq q, p, q < m$, then $SATD_i$ is called a division of value domain of subjects' *ith* attribute and $Dom(SA_i)_j$ denotes the jth branch.

Secondly, based on Definition 1, we can define a subject group and its number *GID* as follows.

Definition 2: Given a set of partition $\{SATD_i | i = 1, \ldots, l\}$, for a subject group SG, if its first attribute value set is $Dom(SA_1)_x, \ldots,$ the *lth* attribute value set is $Dom(SA_l)_z$, where $Dom(SA_1)_x \in SATD_1, \cdots, Dom(SA_l)_z \in SATD_l$, then the subject group can be expressed as $SG = SG_{x,y,\ldots,z}$ and the group number *GID* can be denoted by $x, y, \ldots, z$.

Further, in a subject group, a subjects' unique group identity *GUID* is defined as follows.

Definition 3: If a subject $s_i = (sa_{i1}, sa_{i2}, \ldots, sa_{il})$ where $sa_{i1} \in Dom(SA_1)_x, sa_{i2} \in Dom(SA_2)_y, \ldots, sa_{il} \in$

$Dom(SA_l)_z$, then the subject $s_i \in SG_{x,y,\ldots,z}$ and i is called the unique group identity $GUID$ of s_i in group $SG_{x,y,\ldots,z}$.

In this paper, a subjects' group number GID and group identity $GUID$ is only decided by the proposed AFG scheme. Thus, we can define the fuzzy identification of a subject as follows.

Definition 4: Suppose that a subject $s = s_i \in SG_{x,y,\ldots,z}$, then s can be uniquely marked with both his group number GID and group identity $GUID$. The tuple (GID, $GUID$), where $GID = x, y, \ldots, z$, $GUID = i$, is called as the fuzzy identification of the subject s.

Without loss of generality, we can define an objects' fuzzy identification similar to the above definitions.

Finally, we can define AFG-based access control policy as follows.

Definition 5: If a subject $s_i \in SG_{x,y,\ldots,z}$ is allowed to use an operation a_j on a object $o_j \in OG_{u,v,\ldots,w}$, then the access control policy can be expressed as

$$pol = \{(s_i, a_k, o_j) | k = 1, \cdots, m; i, j, m \in N\} \quad (1)$$

where $s_i \in SG_{x,y,\ldots,z}$ and $o_j \in OG_{u,v,\ldots,w}$ embody entities' fuzzy identification.

In this paper, the above access control policy is produced at the data creation stage and is used at the data access stage. Based on the above policy, users can access cloud data without concealing their own identity and attribute information.

IV. DESIGN OF PRIGUARDER

A. OVERVIEW

PriGuarder spans the three stages of a cloud service, i.e., *user registration, data creation* and *data access*. In each stage, users are allowed to interact directly or indirectly. In indirect mode, a scheme based on attribute fuzzy grouping is introduced to protect users' identity and attribute privacy. The concrete process is shown in Fig. 2.

In *user registration* stage, users can choose a direct or anonymous registration mode. In the anonymous registration mode, TTP is responsible for converting a user identity based on AFG scheme and sends the converting result (the user's fuzzy identification) to CSPs. With the help of users' fuzzy identification list, CSPs can realize anonymous access control.

In *data creation* stage, users can choose to create data directly or anonymously, and can set the access mode to their data such as direct access or anonymous access.

For direct creation mode, when data owners hope their data to be accessed directly, they initiate the request of data creation, submit their data and a statement on data access rights to CSPs. CSPs generate the access control policy based on data owners' statement on access rights. When data owners hope their data accessed anonymously, they submit a statement on data access rights to TTP. TTP converts this statement into an AFG-based access control policy and sends the policy with data to CSPs.

For anonymous creation mode, when data owners hope their data to be accessed anonymously, they initiate the request of data creation, submit their data and a statement on data access rights to TTP. TTP stores data, converts the statement into an AFG-based access control policy. And then TTP transmits the request, sends data and the AFG-based policy with data to CSPs. When data owners allow their data to be accessed directly or real-namely, they submit their data and a statement on data access rights to TTP. TTP transmits the request, sends data and the statement to CSPs. CSPs generate the access control policy of data based on data owners' statement.

In *data access* stage, user can choose to access data directly or anonymously. In anonymous access mode, TTP converts user identity in the request based on AFG scheme and forwards the new request to CSPs. CSPs respond to the user's request.

To sum up, all above three stages need the AFG scheme stated in section IV.B. In addition, exploiting data encryption and timestamp techniques at data creation and data access stages, new access control protocols are proposed to regulate direct and indirect interactions between users and CSPs. The protocols will be stated in section IV.C.

B. AFG SCHEME

In *PriGuarder*, the AFG scheme is the key to protect user privacy, including those from their data, identity and attributes. The AFG scheme introduces a new kind of attribute-based grouping rule. The grouping concept has appeared in group encryption scheme [28], where it produces a common group key which can only be used by group members to decrypt a sharing information. But the AFG scheme is completely different with [28]. The goal of attribute-based grouping is to ensure CSPs have no information about cloud users' attributes during the three stages of a cloud service, i.e., user registration, data creation and data access. First, an attribute value can be conversed as an ASCII tuple. Second, an attribute group number is computed and assigned to the attribute value through linear or nonlinear operation on its ASCII tuple under the control of an operational factor Key. Finally, a users' fuzzy identification is achieved by aggregating all the attribute group numbers of his attribute values.

To be convenient for explanation, $f(S_{Attr})$ is unified to express the grouping rule of subject attribute S_{Attr} later. That means, if a subject $s = (sa_1, sa_2, \ldots, sa_l) \in S_{Attr}$, $f(s) = f((sa_1, sa_2, \ldots, sa_l)) = GID$ where $GID = x, y, \ldots, z$ is defined in Definition 4. $f(S_{Attr})$ works as follows.

1) ASCII CONVERSION

For each attribute value $sa_i (i = 1, \ldots, l, l \in N)$, sa_i is represented as a character string tuple with n_i characters as a group and denoted by $(a_1, a_2, \ldots, a_j)$, where if the final character string a_j is less than n_i and then its tail up to 0. Furthermore, $(a_1, a_2, \ldots, a_j)$ is conversed as an ASCII tuple $(b_1, b_2, \ldots, b_j)$ where b_t $(t = 1, \ldots, j)$ is an ASCII matched witha_t $(t = 1, \ldots, j)$.

For example, suppose that a user's value of Country Attribute is *American*, the attribute value can be expressed as (*Ameri*, *can*00) with 5 characters as a group. And then the

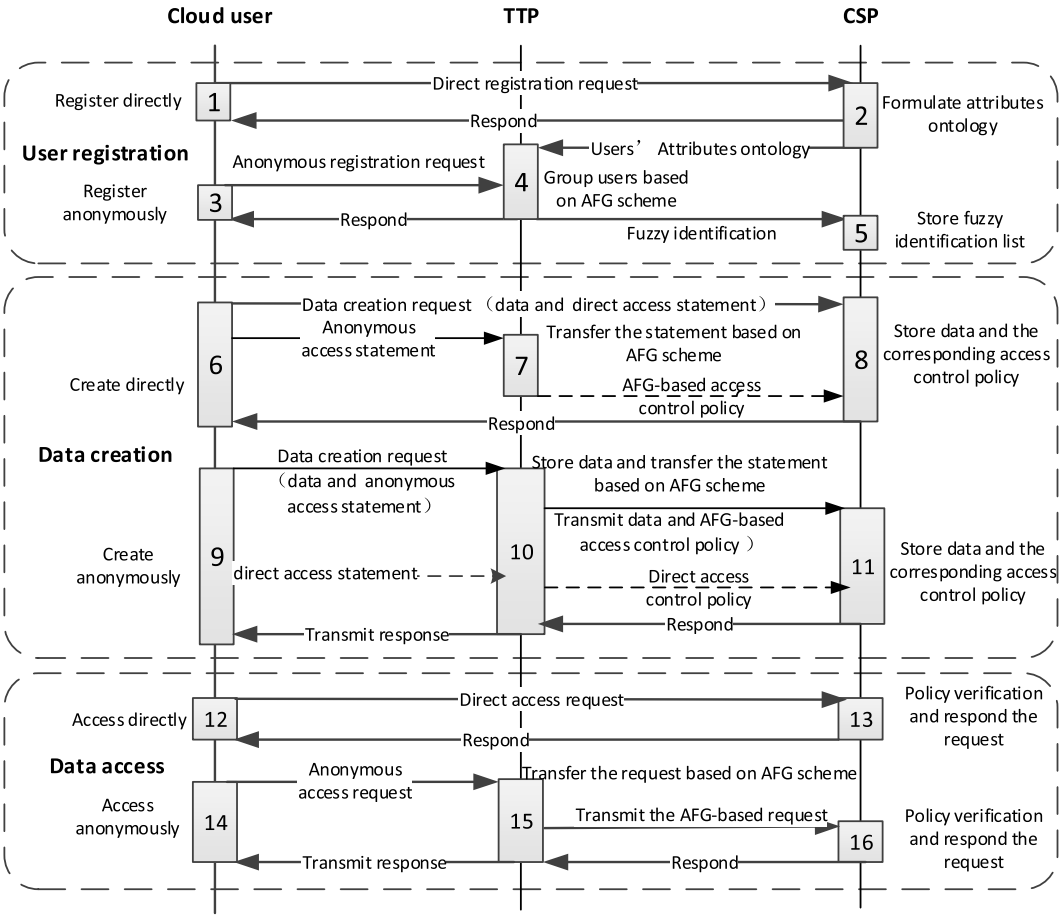

FIGURE 2. The basic principle of *PriGuarder*.

character string tuple (*Ameri*, *can*00) can be conversed as an ASCII tuple (65110101114105, 99971104848).

2) DOUBLE MODULO OPERATORS

For each attribute value sa_i ($i = 1, \ldots, l,\ l \in N$), the attribute group number g_i of sa_i is computed as the following formula.

$$g_i = (b_1\%Key + b_2\%Key + \cdots + b_j\%Key)\%W_i \qquad (2)$$

where $(b_1, b_2, \ldots, b_j)$ is the ASCII tuple of sa_i, Key is the operational factor, W_i is the expected group number of the ith subject attribute type and $\%$ is the *modulo* operator. Thus, all attribute values with the same g_i can constitute an attribute group. As stated in Definition 1, the group should be a division branch of subjects' *ith* attribute value domain. In order to improve the computational complexity of $f(S_{Attr})$, we take prime numbers as Key. It should be pointed out that there may be different n_i and W_i for different attribute type i in $f(S_{Attr})$.

3) GENERATION OF GID

The GID of a subject s $GID = x, y, \ldots, z$ can be generated by combining all g_i ($i = 1, \ldots, l$) together, where $g_1 = x$, $g_2 = y, \ldots, g_l = z$.

In order to protect users' attribute privacy, we should make it difficult for attackers to deduce the exact attribute value of users from their GIDs. Therefore, we introduce two parameters complexity threshold S and security threshold$P$ into AFG scheme.

Complexity threshold S is used to improve the difficulty of finding an attribute value matched with user in an attribute group. Suppose that a users' group number is $GID = x, y, \ldots, z$, where the numbers of attribute values in the corresponding attribute group are $q_x, q_y, \ldots, q_z$ respectively, then the largest matching times of finding the users' all attribute values is $N = \sum_{i=x}^{z} q_i$ and the computation complexity $O(N) > S$. Usually, the threshold S is set on the performance of machine used to matching calculation. For instance, if a computer can match 10^8 times per second, then it will match 8.64×10^{12} per day. At this time, Sshould be greater than 8.64×10^{12}.

Security threshold P is mainly used to control the entity proportion whose attributes will not be leaked in Cloud environment. After $f(S_{Attr})$ ends, users with the same GID can form an entity group. Suppose that there are Rentity groups in the whole Cloud system. In an entity group, CSP has the same computation complexity of matching any

TABLE 1. Attribute Fuzzy Grouping Algorithm.

Input： TTP receives users' Attributes ontology S_{Attr} and $Dom(SA_i)$ send by CSP Output： *GID* and *GUID*
1、 For any $(sa_1,sa_2,\ldots, sa_l) \in S_{Attr}$, TTP formulates the expected group number of the *i*th subject attribute type W_i $(i=1,\ldots,l)$ and *Key*, computes $STAD_i$ of $Dom(SA_i)$ on S_{Attr} by $f(S_{Attr})$, where $f(S_{Attr})$ includes *ASCII conversion, Double modulo operators* and *Generation of GID* stated in above.
2、 For the user $s=(sa_1,sa_2,\ldots, sa_l)$, TTP computes the largest matching times of finding the users' all attribute values *N*. If the attribute group with sa_i has q_i attribute value, then $N=\sum_{i=1}^{l} q_i$.
3、 TTP do sample testing to *R* entity groups with the same *GID*, and compare the $O(N)$ with *S*. If there are *Q* entity groups with $O(N)>S$ and there is $Q/R > P$, then execute Step 5; Otherwise, execute Step 4. As stated in the above, S is complexity threshold and *P* is security threshold set by TTP.
4、 TTP changes *Key* , decreases W_i and does new $f(S_{Attr})$ operations to obtain $STAD_i$, execute Step 3.
5、 TTP distributes user's *GID* and *GUID* according to Definition 4. In this paper, *GUID* is computed by random number generating function in Java, that means, take a random integer between 1 and *a* by the formula *ROUND*(*RAND*()**a*)+1.

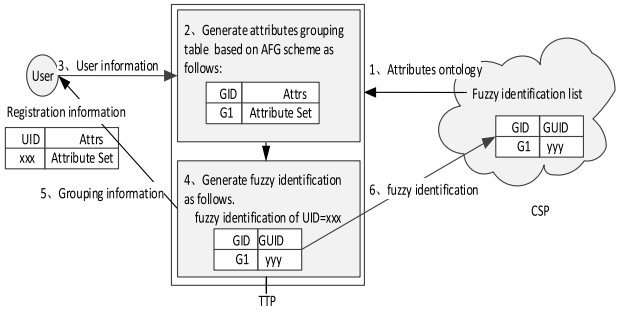

FIGURE 3. AFG-based user's entity conversion.

user and his attribute values on the premise of knowing GID. Let $O(N) > S$ be the computation complexity. If there are Q entity groups where their computation complexity $O(N) > S$, then Q/R reflects the entity proportion where attributes will not be leaked in the whole Cloud system. Obviously, we expect $Q/R > P$ where P is a percentage such as 95%. If $Q/R < P$, it needs to make more entity groups with $O(N) > S$. Based on the above introduction, it must reduce W_i to increase q_i. Thus, it need to reset $f(S_{Attr})$ until $Q/R > P$ meets.

The attribute fuzzy grouping algorithm is shown in Table 1.

In *PriGuarder*, the main function of AFG scheme embodies as follows. First, entity identity is converted into a fuzzy identification. Second, a general access control policy is converted into AFG-based access control policy by using fuzzy identification to replace subject entity in anonymous access statement. Third, a common user request is converted into an AFG-based request by replacing subject entity in the request with fuzzy identification. Here takes user's entity conversion as an example, the process is shown in Fig. 3.

C. INTERACTIVE PROTOCOL OF ACCESS OF ACCESS CONTROL

Since cloud users are allowed to choose different data creation and access modes, there are different access interactive protocols in cloud. The protocols are provided in this section and used to regulate the interaction among users, TTP and CSP. Here we first give the explanation of symbols used in the protocols, as shown in Table 2.

TABLE 2. Symbols used in interactive protocols of access control.

Symbol	Meaning
MSG	Data file
Owner, Reader, Writer	The owner, reader and writer of MSG
KDCs	Several key distribution centers
SKs	Keys used to encrypt and decrypt MSG, distributed by KDCs
Access statement (denoted by X)	A statement on data access rights. Include direct and anonymous access statement.
X_autonym	Direct access statement, used to specify operation right to MSG and identity of authorized users. The function is controlled by {0, 1} switch. When the switch is set to 0, the statement is null; and when the switch is set to 1, data owner can define direct access statement.
X_anonymity	Anonymous access statement, used to specify operation right to MSG and attributes set authorized users should have. The function is also controlled by {0, 1} switch. When the switch is set to 0, the protocol is null; and when the switch is set to 1, data owner can define anonymous access statement.
Identity Claim(denoted by I)	Users' Identity, used to sign on MSG. In anonymous mode, the Identity is fuzzy identification R.
C_o 、C_r、C_w	Permission certificate issued by the cloud or a TTP, used to obtain SKs by users.
R	Fuzzy identification of user, converted by the TTP and used to sign the MSG.
C	Cipher text of MSG encrypted by SKs. Whether or not user access the cipher text depends on whether his identity or fuzzy identification accord with the Access statement or policy.
Grouping policy	User grouping policy formulated by TTP, including the group number GID and the unique group identity GUID in GID.
t	Timestamp, used to prevent replay attacks.

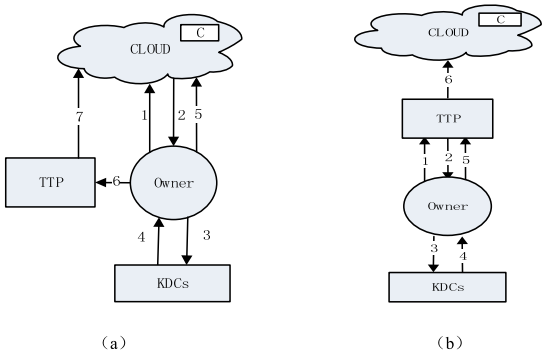

FIGURE 4. Owner's interactive protocol in data creation stage. (a) Owner create data directly. (b) Owner create data anonymously.

1) DATA CREATION STAGE

a. When Owner create data directly, the interaction is shown in Fig. 4 (a).

1. Owner sends a request for creating data MSG in Cloud;
2. Cloud verifies Owner and issue him a certificate C_o;
3. Owner sends C_o to KDCs for requesting SKs;
4. KDCs returns the requested SKs to Owner;
5. Owner signs MSG with Identity Claim, encrypts it by SKs, and gets the cipher text C, and then sends C and X to the Cloud. In default, X_autonym switch in X is set to 1. CSP in Cloud verifies Owner's timestamp t, if passed, then remembers C and generates access control policy of data MSG according to the X_autonym.
6. If Owner allows other users to access his data anonymously, X_anonymity switch in X is set to be 1 and then X_anonymity is sent separately to the TTP.
7. TTP converts X_anonymity to an AFG-based access control policy and sent it to Cloud.

b. When Owner create data anonymously, the interaction is shown in Fig. 4 (b).

1. Owner sends a request to the TTP for creating MSG;
2. TTP gives Owner a C_o according to the Owners' grouping results R, and uses C_o as Owners' proof to ask SKs from KDCs;
3. Owner sends C_o to KDCs;
4. KDCs returns the requested SKs to Owner;
5. Owner signs the MSG with R, encrypts it by SKs and gets cipher text C, and then sends C and X to the TTP. In default, X_anonymity switch in X is set to 1. At this point, if Owner allows users to access data MSG directly, X_autonym switch in X must be set to 1, and then sends X_autonym to the TTP.
6. TTP converts X_anonymity to an AFG-based access control policy and send C and X to Cloud. Cloud verifies Owner's R and t, if passed, he stores C and the AFG-based access control policy in X, and also generates and stores the corresponding access control policy based on X_autonym.

2) DATA ACCESS STAGE

a. When Reader/Writer access data directly, the interaction is shown in Fig. 5 (a)

1. Reader/Writer sends a read/write MSG request to the Cloud;
2. Cloud verifies the user's identity according to the access control policy generated by X_autonym. If passed, he sends C_r/C_w and C to the user;
3. Reader/Writer sends C_r/C_w to KDCs to get SKs;
4. Reader/Writer obtains SKs, then uses them decrypt C and read/write MSG;

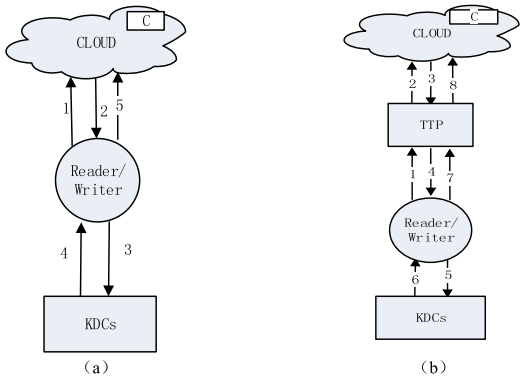

FIGURE 5. Reader/Writer's interactive protocol in data access stage. (a) Reader/Writer access data directly. (b) Reader/Writer access data anonymously.

5. After Writer has written MSG, he signs MSG with his Identity Claim, encrypts it with initial Owner's SKs and gets a new cipher text C. Then he sends C and the timestamp t to Cloud. Since Writer has passed the verification in Step 2, Cloud stores cipher text C and t.

b. When Reader/Writer access data anonymously, the interaction is shown in Fig. 5 (b)

1. Reader/Writer sends a read/write request to TTP;
2. TTP transfers the request by replacing access subject identity with Reader/Writers' fuzzy identification, and sends the transferred request to the Cloud;
3. Cloud uses MSG's AFG-based policy to verify the transferred request, and decides Reader/Writer's permissions, if passed, he sends C_r/C_w to TTP;
4. TTP forwards C_r/C_w to Reader/Writer;
5. Reader/Writer sends C_r/C_w to KDCs for requesting SKs;
6. Reader/Writer receives SKs, then decrypts C by SKs and get data MSG;
7. After Writer has written MSG, he signs MSG with his fuzzy identification and encrypts it with initial Owner's SKs and gets a new cipher text C. Then he sends the C and the timestamp t to the TTP;
8. TTP transfers the request based on Writer's fuzzy identification, puts forward the new request, C and t to Cloud. Cloud has verified Writer's fuzzy identification R in Step 2, Cloud stores cipher text C and t.

V. ILLUSTRATIVE EXAMPLE

In this section, we provide an illustrative example based on the Amazon S3 service. Amazon S3 service doesn't support anonymous access currently. Before users buy the service, they need to fill in the registration information on 10 attributes types, including email, user name, full name, company name,

country, address, nationality, Zip, Phone, VISA card number. Without loss of generality, we select five attribute types including user name, email, nationality, province and VISA card number as an example to illustrate the AFG scheme as follows.

a. For name, email, nation and province attribute, each attribute value is represented as a character string tuple with 5 characters as a group. And for VISA card number attribute, each attribute value is represented as a character string tuple with 8 numeric characters as a group.

b. For $f(S_{Attr})$, the expected group number W_i of name, email, VISA card number, nations, and provinces are 111, 111, 111, 5 and 20 respectively.

c. Set the operational factor $Key = 171$.

d. Suppose that a computer can match 10^8 times per second, then it will match 8.64×10^{12} times per day. Hence, let the complexity threshold $S = 10^{15}$ and the security threshold $P = 98\%$.

e. GUIDs are computed by the following formula

$$\text{ROUND (RAND()*1000)} + 1.$$

According to Table 1, we check the above attribute grouping rule $f(S_{Attr})$. After calculation, the numbers of attribute values in the attribute groups of name, email, nations, VISA card number and provinces are 900, 900, 38, 900, 50 respectively.

The other five attribute types have similar to the above calculation. The numbers of attribute values for full name, address, post code, telephone and Company name are 900,900,50,900,50. Then

$$N = \sum_{i=1}^{10} q_i = 900^6 \times 50^2 \times 38 \times 50 = 2.52434475 \times 10^{24} > S.$$

The entity set proportion $Q/R = 1 > P$. Hence, $f(S_{Attr})$ is feasible and can be used to convert entity identity, access control policy and access request.

A. USER REGISTRATION

Now we use the above scheme to group the anonymous registration users. Suppose that there are three users with anonymous registration request. Their attribute values are shown in Table 3. According to the AFG method, the three users' attribute group numbers are shown in Table 4. According to Definition 4, it is easy to get $Alice \in SG_{102,56,2,8,91}$, $Tom \in SG_{66,20,1,9,6}$, Lucy $\in SG_{56,30,2,17,103}$. And then their fuzzy identifications are shown in Table 5.

B. DATA CREATION

For a more intuitive representation, here we take an example, where Alice create data MSG1 directly and define data access statement which allows both accesses directly and anonymously. The statement is shown in Table 6. In the direct data creation mode, Alice needs to send data MSG1 and X_autonym in the above statement to the cloud directly, then the cloud stores MSG1 and generates the direct access control policy based on X_autonym; Meanwhile, Alice needs to send X_anonymity to TTP, then TTP generates the AFG-based access control policy according to X_anonymity. The AFG-based access control policy on MSG1 is shown in Table 7.

TABLE 3. Users' attribute values.

User Name	Email	Nation	Province	VISA card number
Alice	Alice@yahoo.com	American	Alaska	4062 5403 0108 2212
Tom	Tom@yahoo.com.cn	China	Beijing	4367 4500 5047 1719
Lucy	Lucy@163.com	American	California	4063 6513 4042 1805

TABLE 4. Users' attributes grouping.

User Name (SA_1)	Email (SA_2)	Nation (SA_3)	Province (SA_4)	VISA card number (SA_5)
102	56	2	8	91
66	20	1	9	6
56	30	2	17	103

TABLE 5. Users' fuzzy identifications.

USER NAME	GID	GUID
ALICE	102,56,2,8,91	786
TOM	66,20,1,9,6	246
LUCY	56,30,2,17,103	149

TABLE 6. Access statement on MSG1 defined by Alice.

Access statement	Permission	Statement content
X_autonym	1 (allow)	User named Alice, Tom has read and write rights User named Lucy has read-only right
X_anonymity	1 (allow)	User named Alice, Tom has read and write rights User named Lucy has read-only right

TABLE 7. AFG-based Access control policy on MSG1.

Access control policy	Specific content
Direct access control policy	Alice, Tom have read and write rights; Lucy has read-only right.
AFG-based access control policy	Fuzzy identification: GID= $SG_{102,56,2,8,91}$, GUID=786 and GID= $SG_{66,20,1,9,6}$,GUID=246 have read and write rights; Fuzzy identification: GID= $SG_{56,30,2,17,103}$, GUID=149 has read-only right.

C. DATA ACCESS

When Alice, Tom, Lucy want to access MSG1 directly, they submit their requests to the CSP. The CSP responds these requests based on X_autonym in Table 6. If Alice, Tom, Lucy want to access MSG1 anonymously, they will submit their requests to TTP. TTP transfers the requests based on their fuzzy identification and sends the converted

requests to the CSP. The cloud responds the request based on X_anonymity in Table 6.

VI. SECURITY ANALYSIS

This section gives a security analysis of *PriGuarder*.

A. USER DATA SECURITY

In *PriGuarder*, user's data MSG is signed by Identity Claim and encrypted by SKs from KDCs into a ciphertext C. and then the ciphertext C is added into a timestamp t and transferred in an encryption communication channel. If an attacker wants to get MSG, he must own SKs but the SKs transmission channel has high security. In addition, the timestamp t is introduced to prevent repeat attacks. Hence, it can improve user data security.

Moreover, there are two kinds of access statement X_autonym and X_anonymity. When the owner uploads ciphertext C, other users need to pass identity authentication from TTP. After the verification, TTP sends credentials to users so that they can obtain SKs to decrypt the ciphertext C. If a user can't be authenticated, he can't decrypt C. It is difficult to learn access protocol and the encryption algorithm, so user data security is guaranteed.

B. USER PRIVACY

In anonymous access process, a CSP only can record the converted fuzzy identification of cloud users and does not know user's other information, so user's identity and attribute leakage caused by curious CSPs can be avoided. We take two security threats into consideration as follows.

1. Under the premise that both the files manipulated by a user and legal users' attributes to access the files are known, an attacker wants to learn single or all attribute values of the user.

In this situation, the following conclusions is obviously drawn. We omit their proofs.

Proposition 1: For $s = (sa_1, sa_2, \ldots, sa_l) \in SG_{x,y,\ldots,z}$, $sa_i \in Dom(SA_i)_m (i \in \{1, \ldots, l\},\ m \in \{x, y, \ldots, z\})$, suppose that there are q_i attribute values in $Dom(SA_i)_m$ and an attacker knows the relationship between an attribute group number and the corresponding attributes values sets. The attacker must match q_i times to learn the users' attribute value sa_i.

Proposition 2: For $s = (sa_1, sa_2, \ldots, sa_l) \in SG_{x,y,\ldots,z}$, $sa_i \in Dom(SA_i)_m (i \in \{1, \ldots, l\},\ m \in \{x, y, \ldots, z\})$, suppose that there are q_i attribute values in $Dom(SA_i)_m$ and an attacker knows the relationship between attribute group numbers and corresponding attributes values sets. The attacker must match $N = \sum_{i=1}^{l} q_i$ times to learn the users' all attribute values and the matching computation complexity $O(N) > S$.

2. Under the premise that one or more of attribute values of a user are known but legal users' attributes to access the files aren't known, an attacker wants to learn the other single or all attribute values of the user.

Proposition 3: For $s = (sa_1, sa_2, \ldots, sa_l) \in SG_{x,y,\ldots,z}$, $sa_i \in Dom(SA_i)_m (i \in \{1, \ldots, l\},\ m \in \{x, y, \ldots, z\})$,, suppose that there are q_i attribute values in $Dom(SA_i)_m$ under the operational factors *Key* and the expected group number W_i. Thus, when an attacker has known the users' group number $SG_{x,y,\ldots,z}$ but not understand the user's all attribute values, the computational complexity that he can learn to sa_i is $O(N(sa_i)) = 10^{j(3n_i-1)}$, where n_i is the string length and j is the number of strings in character string tuple of sa_i as stated in Section IV.B.

Proofs: For each attribute value sa_i $(i = 1, l, l,\ l \in N)$, as stated in Section IV.B, its ASCII tuple is $(b_1, b_2, \ldots, b_j)$ after *ASCII conversion* with n_i. After *Double modulo operators*, the attribute group number g_i of sa_i is computed as the above formula (2). g_i is derived as follows.

$$\begin{aligned} p \times W_i + g_i &= y_1 + y_2 + \cdots y_j \\ b_1 &= p_1 \times key + y_1 \\ b_2 &= p_2 \times key + y_2 \\ &\cdots \\ b_j &= p_j \times key + y_j \end{aligned}$$

Where $p \in N,\ 0 < p < key/W_i$ and $p_1, \ldots, p_j \in N,\ 0 < p_1, \ldots, p_j < h/key < f/key$.

Here f is the maximum ASCII number among the character string tuples of all sa_i. Since the character 'Z' with ASCII code 122, f equals to 122122122 . . . 122, with n_i 122. In order to simplify the calculation, let $h = 10^{3n_i-1} < f$.

Next count the number of possible values of b_t, in fact, it needs calculate the values of p_t and y_t $(t = 1, \ldots, j)$, when $t = 1, b_1 = p_1 \times key + y_1 = p_1 \times key + p \times W_i + g_i$, the number of possible values of b_1 is

$$N(b_1) = \max\left(p_1 \times p \times C_{g_i+1-1}^{1-1}\right).$$

$t = 2, b_1 = p_1 \times key + y_1 = p_1 \times key + p \times W_i + g_i - y_2$, the number of possible values of b_1 is

$$N(b_1) = \max\left(p_1 \times p \times C_{g_i+2-1}^{2-1}\right).$$

Continues.

$t = j, b_1 = p_1 \times key + y_1 = p_1 \times key + p \times W_i + g_i - y_2 - \cdots y_j$, the number of possible values of b_1 is

$$N(b_1) = \max\left(p_1 \times p \times C_{g_i+j-1}^{j-1}\right);$$

Then the number values b_1 is

$$\begin{aligned} N(b_1) = \max\left(p_1 \times p \times C_{g_i+j-1}^{j-1}\right) &= \frac{f}{key} \times \frac{key}{W_i} \times C_{g_i+j-1}^{j-1} \\ &= \frac{f}{W_i} \times C_{g_i+j-1}^{j-1} > \frac{h}{W_i} \times C_{g_i+j-1}^{j-1} \\ &= \left(\frac{10^{3n_i-1} \times c_{g_i+j-1}^{j-1}}{W_i}\right)^j \end{aligned}$$

For $\{b_1, b_2, \ldots, b_j\}$, there are the same character number, that means $N(b_1) = N(b_2) = \ldots = N(b_j)$, thus the matching times of sa_i is

$$N(sa_i) = [N(b_1)]^j > \left(\frac{10^{3n_i-1} \times c_{g_i+j-1}^{j-1}}{W_i}\right)^j$$

Hence, the computational complexity is $O(N(sa_i)) = 10^{j(3n_i-1)}$.

Proposition 4: For $s = (sa_1, sa_2, \ldots, sa_l) \in SG_{x,y,\ldots,z}$, $sa_i \in Dom(SA_i)_m (i \in \{1, \ldots, l\},\ m \in \{x, y, \ldots, z\})$, suppose that there are q_i attribute values in $Dom(SA_i)_m$ under the operational factors *Key* and the expected group number W_i. Thus, when an attacker has known the users' group number $SG_{x,y,\ldots,z}$ but not understand the user's all attribute values, the computational complexity that he can learn to the user's all attribute values is $O(N) = 10^{lj(3n_i-1)}$, where n_i is the string length and j is the number of strings in character string tuple of sa_i.

Proofs: Similar to Proposition 3.

Based on the above propositions, it is easy to draw the following conclusions.

Theorem 1: If only $SG_{x,y,\ldots,z}$ and $Dom(SA_i)_m$ are known, all attribute values that a user has cannot be deduced.

Theorem 2: If only $SG_{x,y,\ldots,z}$ is known and the computational complexity of learning a user's all attribute values is $10^{3lj(n_i-1)}$, all attribute values that a user has cannot be deduced.

Theorem 2 illustrates that the more string length n_i and number of strings j an attribute value contains, the greater computational complexity of learning a user's all attribute values is, and then the more secure user identity or attribute privacy is.

C. CORRELATION SECURITY BETWEEN USER DATA AND IDENTITY

As mentioned in the above, a CSP processes transaction only based on a user's fuzzy identification in anonymous access mode. When a user accesses his data, the CSP cannot obtain the user's real identity and attribute information. Hence, it can cut off the relevance between users' data and his identity in Cloud system. It can prevent malicious CSPs or others from peeking users' operation and doing any malicious things.

VII. EVALUATION

We compare *PriGuarder* with the existing access control methods and conduct some comprehensive simulations to test the performance of the proposed method.

A. FUNCTION ANALYSIS

We have compared *PriGuarder* with other existing access control technologies on functions, including fine-grained, centralized or decentralized, privacy protection and replay attack prevented etc. As listed in Table 8, *PriGuarder* can support flexible, fine-grained and distributed cloud services access control. The added timestamp can enhance network communication security and prevent malicious replay attacks. The AFG scheme can protect users' identity and attribute privacy.

B. EXPERIMENTS

Because the AFG scheme is the core of *PriGuarder*, we test the performance stability of the AFG scheme.

TABLE 8. Function comparison PriGuarder with other existing access control methods.

The literature	Fine-grained or not	Centralized \Decentralized	Type	User attributes hided	Replay attacks prevented
[14]	No	Decentralized	RBAC	No	No
[15]	Yes	Decentralized	ABAC	No	No
[16]	Yes	Decentralized	ABAC	No	Yes
[17]	Yes	Decentralized	ABAC	No	Yes
[19]	Yes	Centralized	ABAC	Yes	No
PriGuarder	Yes	Decentralized	ABAC	Yes	Yes

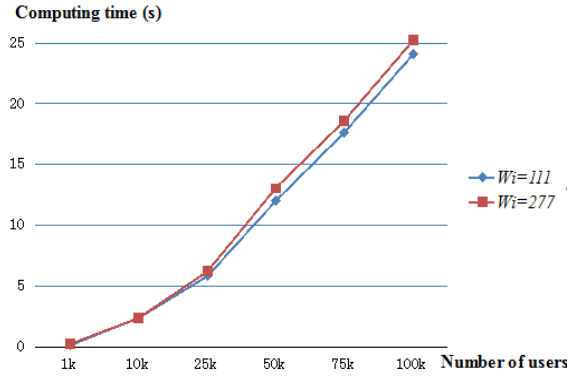

FIGURE 6. Different number of users Vs. the computing time of AFG scheme under different W_i and *key*=171.

1) EXPERIMENTAL ENVIRONMENT

We have implemented the AFG scheme with Python and deployed it in the machine with configuration of Intel (R) Core (TM) i5-3337 - u CPU@1.80GHz processor. We finished the experiments in the example stated in Section V. That means, each user has 10 attribute types including email, user name, full name, company name, country, address, nationality, Zip, Phone, VISA card number.

2) PERFORMANCE OF AFG SCHEME

In the first experiment, let the operational factor $key = 171$ and each attribute's expected group number W_i be 111 and 277 respectively. We count the computing time of AFG scheme under different expected group number W_i. As shown in Fig.6, when the *key* is constant, with the increasing of W_i, the computing time of AFG scheme has a slow increasing trend.

In the second experiment, let each attribute's expected group number $W_i = 171$ and the operational factor *key* be 171 and 87 respectively. We count the computing time of AFG scheme under different operational factors *key*. As shown in Fig. 7, when the W_i is constant, with the increasing of *key*, the computing time of AFG method has a slow decreasing trend.

In summary, with the increasing number of users, the computation time of the AFG scheme increases. Meanwhile, the changes of operational factor *key* and each attribute's expect grouping number W_i have a little influence on the

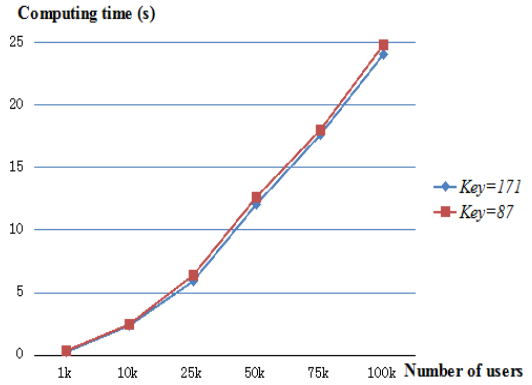

FIGURE 7. Different number of users Vs. the computing time of AFG scheme under different *key* and W_i=171.

whole attribute grouping algorithm. It shows that the AFG scheme has good performance stability.

We haven't compared the experimental performance with other work, i.e., group encryption scheme stated in [28]. Although the proposed AFG scheme and group encryption schemes both use grouping concepts, but their goals are completely different. Group encryption scheme is usually used to solve the information security sharing problem in a group. Its goal is to generate a public group key that is only known by group members. However, our goal is to protect attribute privacy of a cloud user from leaking in three stages of a cloud service. To do this, an attribute value of a cloud user is grouped based on its string format, the fuzzy identification for a cloud user is constructed based on his attribute group number *GID*. In a sense, *GID* can be considered a cipher text for attributes value of a cloud user. The above theoretical analysis has shown that a cloud user's attributes cannot be derived from the cipher text *GID* in proposed AFG scheme.

VIII. CONCLUSION AND FUTURE WORK

To ensure data privacy, we propose *PriGuarder* which spans the three stages of a cloud service, i.e., user registration, data creation and data access. At each stage, users can choose two modes to interact with CSP, i.e., direct or indirect. With the indirect mode, an Attribute Fuzzy Grouping scheme is introduced to ensure user identity privacy and attribute privacy in all the three stages. Furthermore, exploiting data encryption and timestamp techniques, new access control protocols are proposed to regulate interactions between users and CSPs.

However, some problems are not covered in this paper. On the one hand, more efforts should be made to support a more complex trust model, which tackles such situations as a malicious TTP that attempts to reveal users' privacy. On the other hand, our ongoing research will study tradeoffs among qualities of cloud service and privacy protection, some of which may be complementary to *PriGuarder*.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions.

REFERENCES

[1] X. Su *et al.*, "Privacy as a service: Protecting the individual in healthcare data processing," *Computer*, vol. 49, no. 11, pp. 49–59, Nov. 2016.
[2] R. Ahuja and S. K. Mohanty, "A scalable attribute-based access control scheme with flexible delegation cum sharing of access privileges for cloud storage," *IEEE Trans. Cloud Comput.*, to be published.
[3] Z.-L. Gao. *Facebook in the 'Cloud'*. Accessed Nov. 15, 2016. [Online]. Available: http://www.enet.com.cn/article/2011/0516/A20110516859068.shtml
[4] P. Fan. *A Review of Top 10 Information Leakage Events in 2012*, (in Chinese). Accessed: Dec. 13, 2017. [Online]. Available: http://cloud.zol.com.cn/ 345/3451967.html
[5] J. Tang, Y. Cui, Q. Li, K. Ren, J.-C. Liu, and R. Buyya, "Ensuring security and privacy preservation for cloud data services," *ACM Comput. Surv.*, vol. 49, no. 1, Jun. 2016, Art. no. 13, doi: http://dx.doi.org/10.1145/2906153
[6] R. Zhang and P. Chen, "A dynamic cryptographic access control scheme in cloud storage services," in *Proc. 8th Int. Conf. Comput. Netw. Technol. (ICCNT)*, vol. 4. Gueongju, South Korea, Aug. 2012, pp. 50–55.
[7] J. Huang, M. Sharaf, and C.-T. Huang, "A hierarchical framework for secure and scalable ehr sharing and access control in multi-cloud," in *Proc. 41st Int. Conf. Parallel Process. Workshops (ICPPW)*, Pittsburgh, PA, USA, Sep. 2012, pp. 279–287.
[8] M. Nabeel and E. Bertino, "Privacy preserving delegated access control in the storage as a service model," in *Proc. IEEE 13th Int. Conf. Inf. Reuse Integr. (IRI)*, Las Vegas, NV, USA, Aug. 2012, pp. 645–652.
[9] K. Yang, X.-H. Jia, K. Ren, and B. Zhang, "DAC-MACS: Effective data access control for multi-authority cloud storage systems," in *Proc. INFOCOM*, Turin, Italy, Apr. 2013, pp. 2895–2903.
[10] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures: Achieving attribute-privacy and collusion resistance," Cryptol. ePrint Arch., Tech. Rep. 2008/328, 2008.
[11] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Topics in Cryptology—CT-RSA* (Lecture Notes in Computer Science), vol. 6558, A. Kiayias, Ed. Berlin, Germany: Springer, 2011, pp. 376–392.
[12] M. Jensen, S. Schäge, and J. Schwenk, "Towards an anonymous access control and accountability scheme for cloud computing," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. (CLOUD)*, Miami, FL, USA, Jul. 2010, pp. 540–541.
[13] S. M. Khan and K. W. Hamlen, "AnonymousCloud: A data ownership privacy provider framework in cloud computing," in *Proc. IEEE 11th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Liverpool, U.K., Jun. 2012, pp. 170–176.
[14] Z. Jia, L. Pang, S.-S. Luo, J.-Y. Zhang, and Y. Xin, "A privacy-preserving access control protocol for database as a service," in *Proc. Int. Conf. Comput. Sci. Service Syst. (CSSS)*, Nanjing, China, Aug. 2012, pp. 849–854.
[15] K. Yang and X.-H. Jia, "Attributed-based access control for multi-authority systems in cloud storage," in *Proc. IEEE 32nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Macau, China, Jun. 2012, pp. 536–545.
[16] S. Ruj, M. Stojmenovic, and A. Nayak, "Decentralized access control with anonymous authentication of data stored in clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 384–394, Feb. 2013.
[17] L. A. Dunning and R. Kresman, "Privacy preserving data sharing with anonymous ID assignment," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 2, pp. 402–413, Feb. 2013.
[18] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2006, pp. 89–98.
[19] Z.-G. Wan, J.-E. Liu, and R.-H. Deng, "HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 743–754, Apr. 2011.
[20] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably secure and efficient bounded ciphertext policy attribute based encryption," in *Proc. 4th Int. Symp. Inf., Comput., Commun. Secur.*, New York, NY, USA, Mar. 2009, pp. 343–352.
[21] M. Wang, Z. Zhang, and C. Chen, "Security analysis of a privacy-preserving decentralized ciphertext-policy attribute-based encryption scheme," *Concurrency Comput. Pract. Exper.*, vol. 28, no. 4, pp. 1237–1245, Mar. 2016.

[22] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. ACM Conf. Comput. Commun. Secur.*, HongKong, Apr. 2011, pp. 121–130.
[23] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," *Inf. Sci.*, vol. 180, no. 13, pp. 2618–2632, Jul. 2010.
[24] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 6632, K. G. Paterson, Ed. Berlin, Germany: Springer, 2011, pp. 568–588.
[25] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. USENIX Secur. Symp.*, vol. 49. Aug. 2011, p. 34.
[26] Y. Jing. *Cloud Computing Has Forced Third-Party Status Monitoring Services*. (in Chinese). Accessed: Dec. 13, 2017. [Online]. Available: http://datacenter.chinabyte.com/475/11125975.shtml
[27] L. Lin, J.-P. Huai, and X.-X. Li, "Attribute-based access control policies composition algebra," *J. Softw.*, vol. 20, no. 2, pp. 403–414, 2009.
[28] S. Tanada, H. Suzuki, K. Naito, and A. Watanabe, "Proposal for secure group communication using encryption technology," in *Proc. 9th Int. Conf. Mobile Comput. Ubiquitous Netw. (ICMU)*, Oct. 2016, pp. 1–6.

LI LIN (M'14) received the B.S. and M.S. degrees in mathematics from Guangxi Normal University, Guangxi, China, in 2001 and 2004, and the Ph.D. degree in computer science from Beihang University in 2009. She is currently an Associate Professor and a Master Tutor with the Faculty of Information Technology, Beijing University of Technology, China. She has undertaken research projects funded by the National Natural Science Foundation of China and the Beijing Natural Science Foundation and has been participating in various research projects supported by the National High-Tech Research and Development Program of China (863 Program), and the National Key Basic Research Program of China (973 Program). Her current research interests include cloud computing, information security, trusted computing and policy-based management for distributed systems. She is a member of the China Computer Federation.

TING-TING LIU received the undergraduate degree in information management and information system from GuangXi University in China in 2012 and the master's degree in computer science from the Beijing University of Technology in China in 2015. She is currently a platform planner of CDN product in Beijing Chinacache Communications Technology Company Ltd. Her current research interests include data security and privacy in cloud computing, access control in cloud environment, and cloud service selection. She has participated in some research projects funded by the National Natural Science Foundation of China and Beijing Natural Science Foundation.

SHUANG LI received the bachelor's degree in computer science and technology from the Henan Polytechnic University in China in 2015. He is currently pursuing the master degree in computer science with the College of Computer Science, Faculty of Information Technology, Beijing University of Technology, Beijing, China. His current research interests include virtualization security, cloud storage security, cloud service selection, access control in cloud computing platforms, privacy analysis of cloud service, and trusted computing technology. He is participating in some research projects funded by the National Natural Science Foundation of China and supported by the National High-Tech Research and Development Program of China (863 Program).

CHATHURA M. SARATHCHANDRA MAGURAWALGE received the B.Sc. degree (Hons.) from the School of Computer Science and Electronic Engineering, University of Essex, U.K., and he has been awarded the University of Essex Scholarship for his Ph.D. He is currently a Research Fellow with the University of Essex. He has been involved on various private, national, E.U., and international projects. He has held several program committee memberships, such as ACM SIGCOMM, IEEE CloudCom, and IFIP Networking. His current research interests fall within the general area of future computing technologies.

SHAN-SHAN TU received the Ph.D. degree from the Computer Science Department, Beijing University of Post and Telecommunication, in 2014. He is currently an Assistant Professor with the Faculty of Information Technology, Beijing University of Technology, China. He was with the Department of Electronic Engineering, Tsinghua University as a Post-Doctoral Researcher from 2014 to 2016. He visited the University of Essex for joint doctoral training from 2013 to 2014. His research interests are in the areas of cloud computing security and information hiding analysis technology.