IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 11, 2016, accepted September 6, 2016, date of publication September 16, 2016, date of current version March 2, 2017.

Digital Object Identifier 10.1109/ACCESS.2016.2609884

Two-Factor Data Access Control With Efficient Revocation for Multi-Authority Cloud Storage Systems

XIAOYU LI[1], SHAOHUA TANG[1], LINGLING XU[1], HUAQUN WANG[2], AND JIE CHEN[3]

[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[2]College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[3]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200241, China

Corresponding author: S. Tang (shtang@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61632013, Grant U1135004, and Grant 61170080, in part by the 973 Program under Grant 2014CB360501, in part by the Guangdong Provincial Natural Science Foundation under Grant 2014A030308006, and in part by the Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2011).

ABSTRACT Attribute-based encryption, especially for ciphertext-policy attribute-based encryption, can fulfill the functionality of fine-grained access control in cloud storage systems. Since users' attributes may be issued by multiple attribute authorities, multi-authority ciphertext-policy attribute-based encryption is an emerging cryptographic primitive for enforcing attribute-based access control on outsourced data. However, most of the existing multi-authority attribute-based systems are either insecure in attribute-level revocation or lack of efficiency in communication overhead and computation cost. In this paper, we propose an attribute-based access control scheme with two-factor protection for multi-authority cloud storage systems. In our proposed scheme, any user can recover the outsourced data if and only if this user holds sufficient attribute secret keys with respect to the access policy and authorization key in regard to the outsourced data. In addition, the proposed scheme enjoys the properties of constant-size ciphertext and small computation cost. Besides supporting the attribute-level revocation, our proposed scheme allows data owner to carry out the user-level revocation. The security analysis, performance comparisons, and experimental results indicate that our proposed scheme is not only secure but also practical.

INDEX TERMS Attribute-based encryption, multi-authority cloud storage, two-factor protection, attribute-level revocation, user-level revocation.

I. INTRODUCTION

As a new computing paradigm, cloud computing has attracted extensive attentions from both academic and IT industry. It can provide low-cost, high-quality, flexible and scalable services to users. In particular, cloud computing realizes the pay-on-demand environment in which various resources are made available to users as they pay for what they need.

Cloud storage is one of the most fundamental services [1], which enables the data owners to host their data in the cloud and through cloud servers to provide the data access to the data consumers (users). However, it is the semi-trusted cloud service providers (CSPs) that maintain and operate the outsourced data in this storage pattern [2], [3]. Therefore, the privacy and security of users' data are the primary obstacles that impede the cloud storage systems from wide adoption [4], [5]. To prevent the unauthorized entities from accessing the sensitive data, an intuitional solution is to encrypt data and then upload the encrypted data into the cloud [6], [7]. Nevertheless, the traditional public key encryption and identity-based encryption (IBE) [8] cannot be directly adopted. The reason is that they only ensure the encrypted data can be decrypted by a single known user, such that it will decrease the flexibility and scalability of data access control.

Attributed-based encryption (ABE) proposed by Sahai and Waters in [9], can be viewed as the generalization of IBE [8]. In an ABE system, each user is ascribed by a set of descriptive attributes. The user's secret key and ciphertext are associated with an access policy or a set of attributes. Decryption is possible if and only if the attributes of ciphertext or secret key satisfy the access policy. Such an advantage

makes ABE simultaneously fulfill the data confidentiality and fine-grained access control in cloud storage systems. Goval et al. [10] formulated two complimentary forms of ABE: key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). In KP-ABE, user's secret key is associated with an access policy and each ciphertext is labeled with a set of attributes; while in CP-ABE, each ciphertext is associated with an access policy and user's secret key is labeled with a set of attributes. Compared with KP-ABE, CP-ABE is more suitable for the cloud-based data access control since it enables the data owner to enforce the access policy on outsourced data.

However, there remains several challenges to the application of CP-ABE in cloud-based data access control. On one hand, there is only one attribute authority (AA) in the system responsible for attribute management and key distribution [11]–[13]. This precondition cannot satisfy the practical requirements once users' attributes are issued by multiple AAs. For example, a studying abroad agency encrypts some specific messages under the access policy (''SCUT.student'' and ''TOEFL=105''). In this way, only the receiver who is the student of SCUT and now has a TOEFL score of 105 can recover these messages. One important thing to note about these two attributes is that the attribute ''SCUT.student'' is administrated by the SCUT.Registry and the attribute ''TOEFL=105'' is issued by the ETS. On the other hand, in most existing schemes, the size of ciphertext linearly grows with the number of attributes involved in the access policy, which may incur a large communication overhead and computation cost. This will limit the usage of resource-constrained users. Last but not the least, the attribute-level revocation [11], [14] is very difficult since each attribute is conceivably shared by multiple users.

A. RELATED WORK

As mentioned above, CP-ABE is a promising cryptographic mechanism for fine-grained access control. Bethencourt et al. explicitly formalized the notion of CP-ABE and proposed a CP-ABE scheme in [15], but its security proof was given in the generic group model. Cheung and Newport [16] proposed another CP-ABE scheme that supports $AND^*_{+,-}$ access policy, and proved its security under decision bilinear Diffie-Hellman assumption. Later, a number of CP-ABE schemes were proposed [17]–[21] for better efficiency, or security, or expressiveness. The first multi-authority ABE (MA-ABE) scheme was proposed by Chase in [22], where there are several AAs and one central authority (CA) in the system. Each AA issues a set of attribute secret keys to each user, while the CA distributes a global unique identifier together with a final secret key to each user. Other multi-authority ABE schemes have been proposed in [23]–[27].

Emura *et al.* [28] put forth a CP-ABE scheme with constant-size ciphertext. And yet, their scheme only supports the (n, n)-threshold access policy on multi-valued attributes. Another CP-ABE scheme with constant-size ciphertext was proposed in [29], and works for the (t, n)-threshold case. Cheng et al. [30] proposed two new CP-ABE schemes, which have both constant-size ciphertext and small computation cost for $AND^*_{+,-}$ access policy. Sreenivasa and Dutta [31] proposed the first fully security CP-ABE scheme with constant-size ciphertext by adopting the technique of [28] over composite order bilinear group.

The revocation issue is an important and cumbersome problem in attribute-based systems. Several CP-ABE schemes which support attribute-level revocation have been proposed in [11], [13], [14], and [32]. For attribute-level revocation, any revoked user only loses part access privileges as some attributes are removed. That is, each revoked user can still access the data as long as his/her remaining attributes satisfy the access policy. Besides binding an expiration time to each attribute, the revocation methods in CP-ABE schemes can be classified into two categories: directly revocation [32] and indirectly revocation [11], [13], [14]. In the direct revocation, the AA publishes the revocation list so that users can integrate revocation information into the ciphertext while encrypting data. A non-revoked user can decrypt the ciphertext only if the attributes of that user satisfy the access policy in the ciphertext. The advantage of this method is that the attribute-level revocation can be enabled without updating attribute secret keys for the non-revoked users. In the indirect revocation, the AA needs to update the secret key with respect to the revoked attribute for each non-revoked user, instead of making the revocation list public to users. Concretely, Zhang et al. [32] drew support from an auxiliary function to indicate which ciphertexts are involved in revocation events to update these involved ciphertexts. Yu et al. [14] proposed a CP-ABE scheme with indirect attribute-level revocation by the semi-trusted proxy deployed in the data server. The key re-randomization is adopted in Yang et al.'s CP-ABE scheme [13]. Hur and Noh [11] proposed an immediate attribute-level revocation mechanism in CP-ABE by utilizing a binary key-encrypted-key tree for attribute group key distribution. Different from the attribute-level revocation, user-level revocation makes the revoked users lose all the access privileges in the system. In [33], Attrapadung and Imai proposed a CP-ABE scheme with direct user-level revocation by combining the techniques of broadcast encryption and ABE.

B. MOTIVATION

A number of CP-ABE schemes with respect to data access control for multi-authority cloud storage systems have been proposed in [2] and [34]–[37]. In order to achieve the revocation functionality, the proposed schemes in [2], [34], and [36] need secure communication channels to update the attribute secret keys for the non-revoked users. But, Wu et al. [35] pointed out that Yang et al's DAC-MACS scheme [2] cannot guarantee the backward security in active attack model. The reason is that any revoked user still retrieves his/her ability to decrypt some confidential data as a non-revoked user when he/she intercepts the ciphertext update keys delivered from the involved AA. The same security weakness also exists in the schemes of [34] and [36]. Subsequently, Wu et al. [35]

proposed a new extensive scheme called NEDAC-MACS based on Yang et al's DAC-MACS scheme [2]. From the efficient point of view, the proposed multi-authority CP-ABE schemes in [2], [34], [35], and [37] do not possess the character of constant-size ciphertext. It is a negative impact on communication overhead and/or computation cost. Beyond that, the data owners in the schemes of [2] and [34]–[37] are voiceless in the permission revocation. Because these schemes only supported attribute-level revocation. It is not conducive to performing the commercial properties of data owner in cloud computing. Therefore, to construct secure, efficient and revocable access control scheme for multi-authority cloud storage systems is still meaningful.

C. OUR CONTRIBUTION

Inspired by [38], we propose TFDAC-MACS, a secure, efficient and revocable data access control scheme with two-factor protection for multi-authority cloud storage systems in this paper. On the whole, our proposed TFDAC-MACS can be considered as a multi-authority CP-ABE scheme with double-level revocation mechanism. Compared with the existing CP-ABE schemes for multi-authority cloud storage systems, our proposed TFDAC-MACS has the following significant features:

1) Our proposed TFDAC-MACS can provide two-factor data encryption protection for multi-authority cloud storage systems. Each user needs to satisfy two requirements when recovering the outsourced data. One is the attributes of this user satisfy the access policy, and the other is this user has the authorization key.

2) The proposed TFDAC-MACS supports the AND_m access policy, and enjoys the properties of constant-size ciphertext and small computation cost. By making use of the server-aided re-encryption technology, the proposed TFDAC-MACS achieves the attribute-level revocation. At the same time, the data owner is allowed to execute the user-level revocation in the proposed TFDAC-MACS.

3) Our proposed TFDAC-MACS is proven secure against chosen plaintext attacks in the random oracle model. Since the users' global unique identifier is used to link the attribute secret keys and authorization key, the proposed TFDAC-MACS can resist the collusion attack. Meanwhile, the forward security and backward security are guaranteed in our proposed TFDAC-MACS. Theoretical analysis and experimental results indicate that the computation costs of encryption, decryption and attribute revocation are efficient.

D. ORGANIZATION

The rest of the paper is organized as follows: in Section II, we briefly introduce some preliminaries used in this paper. The overview of our proposed TFDAC-MACS is presented in Section III and the detailed construction is given in Section IV. We analyze the proposed scheme in the terms of both security and performance in Section V. Finally, the conclusion is summarized in Section VI.

II. PRELIMINARIES

In this section, we first review some cryptographic background. Then we introduce the access policies involved in this paper.

A. BILINEAR MAPS AND COMPLEXITY ASSUMPTIONS

Let G, and G_T be two cyclic multiplicative groups with the same prime order p, and g be a generator of G. A bilinear pairing $e : G \times G \rightarrow G_T$ is a bilinear map if it has the following properties:

- Bilinearity: $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for $\forall g_1, g_2 \in G$ and $\forall a, b \in Z_p^*$.
- Non-degeneracy: $e(g, g) \neq 1$.
- Computability: There exists an efficient algorithm to compute $e(g_1, g_2)$ for $\forall g_1, g_2 \in G$.

Using the above notations, the decisional Bilinear Diffie-Hellman (DBDH) assumption [8] and decisional n-Bilinear Diffie-Hellman Exponent (BDHE) assumption [39] are defined as follows.

Definition 1: Given $g^a, g^b, g^c \in G$ for unknown $a, b, c \in Z_p^*$ and a random element $Z \in G_T$, the DBDH problem is to distinguish $e(g, g)^{abc}$ and Z. For an adversary $\mathcal{B}$, define its advantage as $Adv_{\mathcal{B}}^{DBDH}(1^\kappa) = |Pr[\mathcal{B}(g, g^a, g^b, g^c, e(g, g)^{abc}) = 1] - Pr[\mathcal{B}(g, g^a, g^b, g^c, Z) = 1]|$. We say that the DBDH assumption holds, if for any probabilistic polynomial-time (PPT) adversary $\mathcal{B}$, the $Adv_{\mathcal{B}}^{DBDH}(1^\kappa)$ is negligible.

Definition 2: Let g and h be two independent generators of G. Denote $\overrightarrow{y}_{g,\alpha,n} = (g_1, g_2, \cdots, g_n, g_{n+2}, \cdots g_{2n}) \in G^{2n-1}$, where $g_i = g^{\alpha^i}$ for unknown $\alpha \in Z_p^*$. The decisional n-BDHE problem is to distinguish $e(g_{n+1}, h)$ and a random choice of $Z \in G_T$. For an adversary $\mathcal{B}$, define its advantage as $Adv_{\mathcal{B}}^{n-BDHE}(1^\kappa) = |\Pr[\mathcal{B}(g, h, \overrightarrow{y}_{g,\alpha,n}, e(g_{n+1}, h)) = 1] - \Pr[\mathcal{B}(g, h, \overrightarrow{y}_{g,\alpha,n}, Z) = 1]|$. We say that the decisional n-BDHE assumption holds, if for any PPT adversary $\mathcal{B}$, the $Adv_{\mathcal{B}}^{n-BDHE}(1^\kappa)$ is negligible.

B. ACCESS POLICY

An access policy W, namely a ciphertext policy in CP-ABE, is a rule that returns either 0 or 1 given a set of attributes L. In our proposed scheme, we only consider the AND_m access policy [28], [40], *i.e.*, *AND*-gate on multi-valued attributes.

Definition 3: Assume that the universe of attributes $U = \{u_1, \cdots, u_n\}$. Let the number of possible values for u_i be n_i and the possible values be indexed as $V_i = \{v_{i,1}, \cdots v_{i,n_i}\}$. Given an attribute list $L = [L_1, \cdots, L_{n'}]$, where $L_i \in V_i$ be an attribute value for u_i, and a ciphertext policy $W = [W_1, \cdots, W_{n'}]$, we say that L satisfies W if and only if $L_i = W_i$, for $\forall i \in \{1, \cdots, n'\}$. The notation $L \vDash W$ is used to represent the fact that L satisfies W, and the case of L does not satisfy W is denoted by $L \nvDash W$.

Although the expressiveness of AND_m access policy is somewhat restricted as compared with the tree-based and

LSSS-realizable access policy, AND_m access policy still remains useful in reality.

III. OVERVIEW

In this section, we first introduce the system model of TFDAC-MACS. Then, we give the framework of our proposed TFDAC-MACS. At last, the security assumptions and threat models are presented.

A. SYSTEM MODEL

As shown in Fig. 1, TFDAC-MACS consists of five kinds of entities: CA, AAs, data owners, users and CSP.

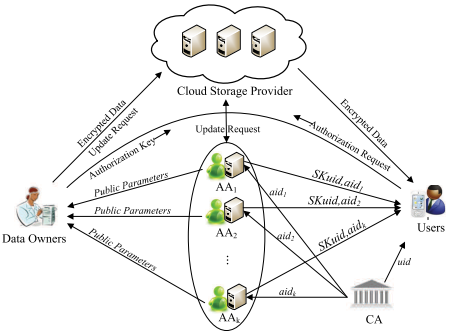


FIGURE 1. System model of TFDAC-MACS.

The CA sets up the system, and responses the registration requests from all the AAs and users. However, the CA is not involved into any attribute-related management.

Each AA administers a distinct attribute domain and generates a pair of public/secret key for each attribute in this attribute domain. Without any doubt, each attribute is only managed by a single AA. Once receiving the request of attribute registration from a user, the AA generates the corresponding attribute secret keys for this user. Additionally, each AA is responsible to execute the attribute revocation of users.

Before uploading a shared data to the cloud storage servers, the data owner defines an access policy and encrypts the data under this access policy. After that, the data owner sends the ciphertext and its corresponding access policy to the CSP. Meanwhile, the data owner is responsible for issuing and revoking the user's authorization.

Each user is labeled with a set of attributes, besides a global unique identifier. In order to obtain the shared data, each user needs to request the attribute secret keys and authorization from AAs and data owner, respectively. Any user can download the ciphertext from the CSP. Only the authorized user who has the specific attributes can successfully recover the outsourced data.

It becomes obvious that the CSP provides data storage service and enforces the process of ciphertext update. The ciphertext update occurs in the following two cases: (1) any of AAs revokes users' one or more attributes; (2) the data owner revokes one or more authorized users.

B. FRAMEWORK OF OUR TFDAC-MACS

The framework of TFDAC-MACS consists of the following phases:

1) PHASE 1 (SYSTEM INITIALIZATION)

First, the CA generates some global public parameters for the system, and accepts both the AA registration and user registration.

Then, each AA and data owner respectively generate the public parameters and secret information used throughout the execution of system.

2) PHASE 2 (SECRET KEY AND AUTHORIZATION GENERATION)

When a user submits a request of attribute registration to AA, the AA distributes the corresponding attribute secret keys to this user if his/her certificate is true. When a user submits an authorization request to data owner, the data owner generates the corresponding authorization key and delivers it to this user.

3) PHASE 3 (DATA ENCRYPTION)

For each shared data, the data owner first defines an access policy, and then encrypts the data under this specified access policy. Thereafter, the data owner outsources this ciphertext to the CSP. The encryption operation will use a set of public keys from the involved AAs and the data owner's authorization secret key.

4) PHASE 4 (DATA DECRYPTION)

All the users in the system are allowed to query and download any interested ciphertexts from the CSP. A user is able to recover the outsourced data, only if this user holds the sufficient attribute secret keys with respect to access policy and authorization key with regard to outsourced data.

5) PHASE 5 (ATTRIBUTE-LEVEL REVOCATION)

For attribute-level revocation, the AA who manages the revoked attribute, issues a new public key to this revoked attribute, and generates attribute update keys for non-revoked users and a set of ciphertext update components for CSP. Each non-revoked user who holds the revoked attribute will update the corresponding attribute secret key upon receiving the attribute update key. Based on the set of ciphertext update components, the ciphertexts associated with the revoked attribute will be updated by the CSP.

6) PHASE 6 (USER-LEVEL REVOCATION)

In order to revoke a user's access privilege, the data owner generates a new authorization secret key used for authorization, a set of authorization update keys for non-revoked users and a set of ciphertext update components for ciphertext update. When receiving the authorization update key, each non-revoked user updates the authorization key and obtains the new version. All the involved ciphertexts will be updated by the CSP based on the set of ciphertext update components.

C. SECURITY ASSUMPTIONS AND THREAT MODELS

Following [2] and [38], we have the following assumptions in TFDAC-MACS:

- The CA is a full trusted party.
- Each AA is also trusted. But, any of AAs will never collude with users.
- The CSP is honest but curious, namely semi-trust. It will correctly execute all the prescribed operations, but may try to decrypt the ciphertexts stored in the cloud servers by itself.
- Each user is dishonest, and may collude with others to obtain unauthorized access to data. Meanwhile, each user is not allowed to expose his/her attribute secret keys and authorization key to an adversary.

Based on the above security assumptions, two threats are considered in this work. One is denoted by **Type-I** threat: decrypt without authorization key, and the other is denoted by **Type-II** threat: decrypt without adequate attribute secret keys. The goal of the adversary in these two threat models is to decrypt the ciphertexts beyond its privileges.

IV. OUR CONSTRUCTION

In this section, we present the detailed construction of our proposed TFDAC-MACS. The main challenge issue is how to make the two factors into an integration, rather than two separate parts of double encryption. We leverage the multi-authority CP-ABE technology to implement the fine-grained access control, and adopt the IBE technology to enhance the security protection. Furthermore, we use the server-aided re-encryption technology to achieve double-level revocation mechanism such that the updated ciphertexts only be recovered by the non-revoked users. The detailed description is presented as follows:

Define the universe of attribute $U = \{u_1, \cdots, u_n\}$. Let $U_{aid} = \{u_{aid_1}, \cdots, u_{aid_{n_{aid}}}\}$ denote the attribute domain for AA_{aid}, where n_{aid} is the total number of attributes managed by AA_{aid}. We set $S_{aid,i} = \{v_{aid_i,1}, \cdots, v_{aid_i,n_i}\}$ be the multi-value set for u_{aid_i} and use the notation $[i]$ to denote the set $\{1, \cdots, i\}$.

A. PHASE 1 (SYSTEM INITIALIZATION)

1) *CASetup*: The CA runs this algorithm to set up the system, which takes the implicit security parameter κ as input.

a) The CA chooses two multiplicative groups G and G_T with the same prime order p. Let g be a generator of G and $e : G \times G \rightarrow G_T$ be a bilinear map. The CA also chooses a hash function $H : \{0, 1\}^* \rightarrow G$. The global public parameters of the system are: $GPP = (p, g, G, G_T, e, H)$.

b) Each AA should register itself to the CA. If the AA is a legal authority in the system, the CA assigns a global unique identifier aid to it.

c) For each user, the CA assigns a global unique identifier uid, a pair of public/secret key (pk_{uid}, sk_{uid}) and a certificate $Cert(uid)$ to this user.

2) *AASetup*: Each AA takes the GPP and its managed attribute domain U_{aid} as the input of this algorithm to generate the public key and its corresponding master secret key.

The AA_{aid} first randomly chooses $x_{aid} \in Z_p^*$ and computes $e(g, g)^{x_{aid}}$. Then, the AA_{aid} randomly chooses $y_{aid_i,j} \in Z_p^*$ and computes $g^{y_{aid_i,j}}$ for each attribute value $v_{aid_i,j} \in S_{aid,i}$. The public key $PK_{aid} = (APK_{aid} = e(g, g)^{x_{aid}}, \{UPK_{aid_i,j} = g^{y_{aid_i,j}} | u_{aid_i} \in U_{aid} \wedge v_{aid_i,j} \in S_{aid,i}\})$, and its corresponding master secret key $SK_{aid} = (ASK_{aid} = x_{aid}, \{USK_{aid_i,j} = y_{aid_i,j} | u_{aid_i} \in U_{aid} \wedge v_{aid_i,j} \in S_{aid,i}\})$.

3) *DOSetup*: The data owner with identifier oid, denoted by DO_{oid}, takes the GPP as the input of this algorithm to generate a pair of public/secret key used for authorization.

The data owner DO_{oid} randomly chooses $\alpha \in Z_p^*$ as the authorization secret key OSK_{oid}, and computes its corresponding public key $OPK_{oid} = g^{\alpha}$.

B. PHASE 2 (SECRET KEY AND AUTHORIZATION GENERATION)

4) *AAKeyGen*: When a user DC_{uid} submits a request of attribute registration to AA_{aid}, the AA_{aid} first authenticates whether this user is a legal user by verifying the certificate $Cert(uid)$. If the $Cert(uid)$ is invalid, it aborts. Otherwise, the AA_{aid} assigns an attribute list $L_{uid,aid}$ to user DC_{uid} according to his/her role or identity. Then, the AA_{aid} runs this algorithm with taking the global public parameters GPP and master secret key SK_{aid} as input. It outputs a set of attribute secret keys $SK_{uid,aid}$ for user DC_{uid}.

Suppose $v_{aid_i,j} \in L_{uid,aid}$, the AA_{aid} computes $SK_{v_{aid_i,j}} = g^{x_{aid_i}} H(uid)^{y_{aid_i,j}}$. Thus, the corresponding attribute secret keys is $SK_{uid,aid} = \{SK_{v_{aid_i,j}} | v_{aid_i,j} \in L_{uid,aid}\}$.

5) *Auth*: When a user DC_{uid} submits an authorization request to data owner DO_{oid}, the data owner DO_{oid} first authenticates whether this user is a legal user by verifying the certificate $Cert(uid)$. If the $Cert(uid)$ is invalid, it aborts. Otherwise, the data owner DO_{oid} runs this algorithm with taking the authorization secret key OSK_{oid} as input. It outputs an authorization key $SK_{uid,oid}$ for user DC_{uid}.

The data owner DO_{oid} computes the authorization key $SK_{uid,oid} = H(uid)^{\alpha}$ and delivers it to user DC_{uid}.

C. PHASE 3 (DATA ENCRYPTION)

6) *Enc*: The data owner DO_{oid} runs this algorithm to encrypt a data m under the access policy W with taking the public keys $(\bigcup_{aid \in I_W^A} APK_{aid}, \bigcup_{v_{aid_i,j} \in W} UPK_{aid_i,j})$ and authorization secret key OSK_{oid} as input, where I_W^A denotes as the index set of involved AAs. Let n_W^{aid} denote the number of involved attributes that managed by AA_{aid} in W.

The data owner DO_{oid} randomly chooses $s \in Z_p^*$ and sets $CT_W = (W, C_1, C_2, C_3)$, where

$$C_1 = m \cdot \Big(\prod_{aid \in I_W^A} e(g, g)^{x_{aid} n_W^{aid}}\Big)^s, \tag{1}$$

$$C_2 = g^s, \tag{2}$$

$$C_3 = \Big(\prod_{v_{aid_i,j} \in W} g^{y_{aid_i,j}}\Big)^{s+\alpha}. \tag{3}$$

Finally, the data owner DO_{oid} selects an unique label ID_W for this data and uploads the (oid, ID_W, CT_W) onto the CSP.

D. PHASE 4 (DATA DECRYPTION)

7) *Dec*: Upon receiving the ciphertext CT_W from the CSP, user DC_{uid} first check whether $\bigcup L_{uid,aid} \vDash W$. If it is true, the user DC_{uid} runs this algorithm with taking his/her global unique identifier uid, the attribute secret key $\bigcup SK_{uid,aid}$ and the authorization key $SK_{uid,oid}$ to decrypt the data m.

Based on the attribute values in W, the user DC_{uid} aggregates the attribute secret keys to generate $SK_W = \prod_{v_{aid_i,j} \in W} SK_{v_{aid_i,j}}$. Then, the user DC_{uid} computes $UPK_W = \prod_{v_{aid_i,j} \in W} UPK_{aid_i,j}$. The data m is recovered as:

$$m = \frac{C_1 \cdot e(H(uid), C_3)}{e(C_2, SK_W) e(SK_{uid,oid}, UPK_W)}. \tag{4}$$

E. PHASE 5 (ATTRIBUTE-LEVEL REVOCATION)

Assume that an attribute of user DC_{uid} is revoked from AA_{aid} and the revoked attribute value is $v_{aid_i,j}$. The involved AA_{aid} first queries the CSP for ciphertext components $\bigcup (oid, ID_W, C_2)$, where $v_{aid_i,j} \in W$. Then, the AA_{aid} generates a ciphertext update component $CUK_{v_{aid_i,j}}^{ID_W}$ for each (oid, ID_W, C_2). Moreover, the AA_{aid} needs to compute an attribute update key $KUK_{v_{aid_i,j}}^{uid'}$ for each non-revoked user $DC_{uid'}$. This phase contains the following three algorithms:

8) *KeyUpdate*: The AA_{aid} runs this algorithm with taking a non-revoked user list NRU, the secret key ASK_{aid}, the master secret key $USK1_{aid_i,j}$, the public key OPK_{oid} and the ciphertext components $\bigcup (oid, ID_W, C_2)$ as input. It outputs a new $UPK_{aid_i,j}$ for $v_{aid_i,j}$, attribute update keys $\bigcup KUK_{v_{aid_i,j}}^{uid'}$ and ciphertext update components $\bigcup CUK_{v_{aid_i,j}}^{ID_W}$.

a) The AA_{aid} randomly chooses $y'_{aid_i,j} \in Z_p^*$ as the new master secret key for the attribute value $v_{aid_i,j}$. Then, the AA_{aid} computes $UPK_{aid_i,j} = g^{y'_{aid_i,j}}$.

b) For each non-revoked user $DC_{uid'} \in NRU$, the AA_{aid} generates an attribute update key $KUK_{v_{aid_i,j}}^{uid'} = H(uid')^{y'_{aid_i,j} - y_{aid_i,j}}$.

c) For each (oid, ID_W, C_2), the AA_{aid} computes the ciphertext update component $CUK_{v_{aid_i,j}}^{ID_W} = (g^s \cdot g^{\alpha})^{y'_{aid_i,j} - y_{aid_i,j}}$.

The AA_{aid} sends $KUK_{v_{aid_i,j}}^{uid'}$ and $\bigcup (oid, ID_W, CUK_{v_{aid_i,j}}^{ID_W})$ to each non-revoked user $DC_{uid'}$ and the CSP, respectively.

9) *SKUpdate*: Upon receiving the attribute update key $KUK_{v_{aid_i,j}}^{uid'}$, the user $DC_{uid'}$ runs this algorithm to update his/her attribute secret key as $SK'_{v_{aid_i,j}} = SK_{v_{aid_i,j}} \cdot KUK_{v_{aid_i,j}}^{uid'} = g^{x_{aid}} \cdot H(uid)^{y_{aid_i,j}} \cdot H(uid)^{y'_{aid_i,j} - y_{aid_i,j}} = g^{x_{aid}} \cdot H(uid)^{y'_{aid_i,j}}$.

10) *CTAUpdate*: The CSP runs this algorithm to update the involved ciphertexts when receiving $\bigcup (oid, ID_W, CUK_{v_{aid_i,j}}^{ID_W})$.

According to the oid and ID_W, the CSP first retrieves the involved ciphertexts component (C_1, C_2, C_3). Then the CSP randomly chooses $r \in Z_p^*$ and computes

$$C'_1 = C_1 \cdot \Big(\prod_{aid \in I_W^A} e(g,g)^{x_{aid} n_W^{aid}} \Big)^r = m \cdot \Big(\prod_{aid \in I_W^A} e(g,g)^{x_{aid} n_W^{aid}} \Big)^{(s+r)}, \tag{5}$$

$$C'_2 = C_2 \cdot g^r = g^{s+r}, \tag{6}$$

$$C'_3 = C_3 \cdot CUK_{v_{aid_i,j}}^{ID_W} \cdot \Big(\prod_{v_{aid_t,j} \in W, v_{aid_t,j} \neq v_{aid_i,j}} g^{y_{aid_t,j}} \Big)^r \cdot g^{y'_{aid_i,j} r} = \Big(\prod_{v_{aid_t,j} \in W, v_{aid_t,j} \neq v_{aid_i,j}} g^{y_{aid_i,j}} \Big)^{(s+\alpha+r)} \cdot (g^{y'_{aid_i,j}})^{(s+\alpha+r)}. \tag{7}$$

F. PHASE 6 (USER-LEVEL REVOCATION)

Suppose that the data owner DO_{oid} wants to revoke the access privilege of user DC_{uid}. The data owner DO_{oid} first chooses a new authorization secret key and computes its corresponding public key. Then, the data owner DO_{oid} generates a new authorization update key for each non-revoked users and a set of ciphertext update components for ciphertext update. This phase contains the following three algorithms.

11) *DAAuthUpdate*: The data owner DO_{oid} runs this algorithm by taking the old authorization secret key OSK_{oid}, the non-revoked user list NRU' and the public parameters from each AA_{aid} as input.

a) The data owner DO_{oid} randomly chooses $\beta \in Z_P^*$ as the new authorization secret key OSK'_{oid}, and computes the corresponding public key $OPK'_{oid} = g^{\beta}$.

b) For each non-revoked user $DC_{uid'} \in NRU'$, the data owner DO_{oid} generates an authorization update key $AUK_{uid',aid} = H(uid')^{\beta - \alpha}$.

c) For each attribute value $v_{aid_i,j}$, the data owner DO_{oid} computes the ciphertext update component $UAU_{aid_i,j} = (UPK_{aid_i,j})^{\beta - \alpha} = (g^{y_{aid_i,j}})^{\beta - \alpha}$.

Finally, the data owner DO_{oid} sends $AUK_{uid',aid}$ and $(oid, \bigcup UAU_{aid_i,j})$ to each non-revoked user and the CSP, respectively.

12) *AuthUpdate*: Upon receiving the new authorization update key, each non-revoked user $DC_{uid'}$ runs this algorithm to update his/her authorization key as: $SK'_{uid',oid} = SK_{uid',oid} \cdot AUK_{uid',aid} = H(uid')^{\beta}$.

13) *CTOUpdate*: The CSP runs this algorithm to update all the outsourced ciphertexts of data owner DO_{oid} when receiving $(oid, \bigcup UAU_{aid_i,j})$.

For each retrieved ciphertext (oid, ID_W, CT_W), the CSP first randomly chooses $r' \in Z_p^*$ and computes

$$C_1' = C_1 \cdot (\prod_{aid \in I_W^A} e(g,g)^{x_{aid} n_W^{aid}})^{r'} = m \cdot (\prod_{aid \in I_W^A} e(g,g)^{x_{aid} n_W^{aid}})^{(s+r')}, \quad (8)$$

$$C_2' = C_2 \cdot g^{r'} = g^{s+r'}, \quad (9)$$

$$C_3' = C_3 \cdot \prod_{v_{aid_i,j} \in W} UAU_{aid_i,j} \cdot (\prod_{v_{aid_i,j} \in W} g^{y_{aid_i,j}})^{r'} = (\prod_{v_{aid_i,j} \in W} g^{y_{aid_i,j}})^{(s+\beta+r')}. \quad (10)$$

Notice that each user in the system can verify the validity of his/her authorization key by $e(OPK_{oid}, H(uid)) \stackrel{?}{=} e(g, SK_{uid,oid})$. If it is true, the authorization key issued by the data owner is valid. Otherwise, the authorization key that this user holds is invalid.

V. SECURITY ANALYSIS AND PERFORMANCE COMPARISON

In this section, we prove that our proposed TFDAC-MACS is secure against the two threats in Section III-C. We also discuss that the proposed TFDAC-MACS can guarantee data confidentiality against the CSP, collusion resistance, forward security and backward security. The performance comparisons and experiment simulations are also carried out in this section.

A. SECURITY ANALYSIS

As we have said, there are two threats in our proposed TFDAC-MACS. Thus, we separate two security levels for the data confidentiality of our proposed scheme in the static cases. One is allowing an adversary to obtain the attribute secret keys but not the authorization key, and the other is the reversed case.

For **Type-I** Security: To prove the security of our proposed TFDAC-MACS for static **Type-I** adversary, we construct an IBE scheme, called VBFIBE. The VBFIBE scheme is a various of Boneh and Franklin's IBE scheme [8]. There are two differences between these two schemes.

- Message space: The VBFIBE scheme requires $M \subseteq G_T$, while Boneh and Franklin's IBE scheme sets $M \subseteq \{0,1\}^n$.
- The form of ciphertext: $C = (C_1 = g^r, C_2 = m \cdot g_{ID}^r)$ and $C = (C_1 = g^r, C_2 = m \oplus H_2(g_{ID}^r))$ are the ciphertext of m in these two schemes, respectively.

Lemma 1: Suppose that the DBDH assumption holds in G, then there is no PPT adversary who can break the security of VBFIBE scheme with non-negligible advantage.

Proof: Suppose that there exists an adversary $\mathcal{A}$, which breaks the above VBFIBE scheme with $Adv_{\mathcal{A}}^{VBFIBE}(1^\kappa) \geq \epsilon$. We can construct an algorithm $\mathcal{B}$ that solves the DBDH problem in G. By taking (g, g^a, g^b, g^c, Z) as input, $\mathcal{B}$'s goal is to output 1 if $Z = e(g,g)^{abc}$ and 0 otherwise. $\mathcal{B}$ works by interacting with $\mathcal{A}$ in a selective identity game [41] as follows:

Initialization: The selective identity game begins with $\mathcal{A}$ first choosing an identity uid^*.

Setup: $\mathcal{B}$ sets $P_{pub} = g^a$, and sends the system parameters of VBFIBE (g, p, P_{pub}) to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ adaptively issues the following queries to $\mathcal{B}$.

- **H_1-query**: $\mathcal{B}$ simulate the random oracle H_1 for $\mathcal{A}$'s H_1 queries by maintaining a table H_1^{list}. When $\mathcal{A}$ submits the identity uid^* to $\mathcal{B}$, $\mathcal{B}$ responds to $\mathcal{A}$ with $Q_{uid^*} = H_1(uid^*) = g^b$. When $\mathcal{A}$ submits an identity $uid \neq uid^*$ to $\mathcal{B}$, $\mathcal{B}$ responds as follows:
  1. If the query uid already appears in H_1^{list} in a tuple $(uid, Q_{uid}, t, coin)$, then $\mathcal{B}$ responds with $Q_{uid} = H_1(uid)$.
  2. Otherwise, $\mathcal{B}$ randomly chooses $coin \in \{0,1\}$ and $t \in Z_p^*$. If $coin = 0$, then $\mathcal{B}$ computes $Q_{uid} = g^t$. If $coin = 1$, then $\mathcal{B}$ computes $Q_{uid} = g^{bt}$. $\mathcal{B}$ add $(uid, Q_{uid}, t, coin)$ into H_1^{list} and sends Q_{uid} to $\mathcal{A}$ as the response of $H_1(uid)$.
- **Private key query**: $\mathcal{A}$ submits $uid \neq uid^*$ to $\mathcal{B}$ for obtaining the private key d_{uid}. $\mathcal{B}$ first retrieves uid in H_1^{list}. If such an item exists in H_1^{list}, $\mathcal{B}$ obtains $(uid, Q_{uid}, t, coin)$ from H_1^{list}. If $coin_{uid} = 0$, $\mathcal{B}$ responds to $\mathcal{A}$ with $d_{uid} = (g^a)^t$. If $coin_{uid} = 1$, $\mathcal{B}$ reports failure and terminates. If there is no such an items in H_1^{list}, $\mathcal{B}$ first runs the operates in H_1-query and obtains $(uid, Q_{uid}, t, coin)$. After that, $\mathcal{B}$ responds to $\mathcal{A}$ or terminates based on the value of $coin$.

Challenge: When $\mathcal{A}$ decides that the Phase 1 is over, $\mathcal{A}$ submits two messages $M_0, M_1 \in G_T$ to $\mathcal{B}$. $\mathcal{B}$ picks a random bit $b^* \in \{0,1\}$ and computes the challenge ciphertext $C^* = (g^c, M_{b^*} \cdot Z)$. Hence, if $Z = e(g,g)^{abc}$ then C^* is a valid encryption of M_{b^*} under the public key Q_{uid^*}. If Z is a random element in G_T, then C^* is independent of b^* in $\mathcal{A}$'s view.

Phase 2: The same as **Phase 1**.

Guess: $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$. If $b' = b^*$, $\mathcal{B}$ outputs 1 in the decisional BDH game to guess that $Z = e(g,g)^{abc}$. Otherwise, it outputs 0 to indicate that Z is a random element in G_T.

Assume that $\mathcal{A}$ makes a total of q_E private key queries in the above game. The probability that $\mathcal{B}$ does not abort in Phase 1 or 2 is $1/2^{q_E}$. Thus, we have $Adv_{\mathcal{B}}^{DBDH}(1^\kappa) \geq \epsilon/2^{q_E}$.

By the DBDH assumption, we know that the above VBFIBE scheme is IND-sID-CPA secure in the random oracle model. This completes the proof of Lemma 1. ■

Theorem 1: Suppose that there is no PPT adversary who can break the security of VBFIBE scheme with non-negligible advantage. then, there is no PPT **Type-I** adversary who can break our TFDAC-MACS with non-negligible advantage.

Proof: Suppose that there exists a **Type-I** adversary $\mathcal{A}_1$, which breaks the proposed scheme with $Adv_{\mathcal{A}_1}^{TFDAC-MACS}(1^\kappa) \geq \epsilon$. We can construct an adversary $\mathcal{A}_2$ that breaks the VBFIBE scheme. Let the challenger $\mathcal{B}$ administrate the VBFIBE scheme and want to attack an

instance of DBDH. Once $\mathcal{A}_1$ launches the interaction with $\mathcal{A}_2$, $\mathcal{A}_2$ submits an interaction request to $\mathcal{B}$ for obtaining the public parameters.

Setup: $\mathcal{A}_2$ takes the public parameter of VBFIBE scheme (p, g, G, G_T, P_{pub}) from $\mathcal{B}$. Firstly, $\mathcal{A}_2$ randomly chooses $x_i \in Z_p^*$ and computes $APK_i = e(g,g)^{x_i}$ for AA_i. Then, $\mathcal{A}_2$ randomly chooses $y_{i_k,j} \in Z_p^*$ for each attribute $v_{i_k,j}$, and computes $UPKx_{i_k,j} = e(g,g)^{y_{i_k,j}}$. $\mathcal{A}_2$ sends $(P_{pub}, \bigcup APK_i, \bigcup UPK_{i_k,j})$ to $\mathcal{A}_1$, where $i_k \in [n], j \in [n_{i_k}]$.

Phase 1: $\mathcal{A}_1$ makes the following query for obtaining the attribute secret keys. $\mathcal{A}_2$ can answer $\mathcal{A}_1$'s query by executing the H_1-query to $\mathcal{B}$.

AAKeyGen query: Suppose $\mathcal{A}_1$ submits an attribute list L_{uid} to $\mathcal{A}_2$ for a query on attribute secret keys, where uid is a user's global unique identifier. $\mathcal{A}_2$ first submits an H_1-query with uid to $\mathcal{B}$ and obtain the response $H_1(uid)$. Then, $\mathcal{A}_2$ computes $SK_{v_{i_k,j}} = g^{x_i} H_1(uid)^{y_{i_k,j}}$ for $v_{i_k,j} \in L_{uid}$. $\mathcal{A}_2$ sends $SK_{L_{uid}} = \{SK_{v_{i_k,j}} | v_{i_k,j} \in L_{uid}\}$ to $\mathcal{A}_1$ as the response of L_{uid}'s query.

Challenge: $\mathcal{A}_1$ submits an access structure $W^* = W_1^*, \cdots, W_m^*$ and two messages $M_0, M_1 \in G_T$ to $\mathcal{A}_2$. $\mathcal{A}_2$ randomly chooses an global unique identifier uid^* and sends uid^* to B for H_1-query, where $L_{uid^*} \vDash W^*$. Then, $\mathcal{A}_2$ sends the (uid^*, M_0, M_1) to $\mathcal{B}$, and is given the challenge ciphertext $CT = (C_1, C_2 = M_b \cdot T)$. $\mathcal{A}_2$ sets $x_W^* = \sum_{aid \in I_{W^*}^A} n_{W^*}^{aid} x_{aid}$ and $y_W^* = \sum_{v_{i_k,j} = W_{i_k}^*} y_{i_k,j}$. Then, $\mathcal{A}_2$ computes the challenge ciphertext for $\mathcal{A}_1$ from CT as: $CT_{W^*} = (W^*, C_1' = C_2 \cdot e(g, C_1)^{x_{W^*}} \cdot e(H_1(uid^*), C_1)^{y_W^*}, C_2' = C_1, C_3 = (P_{pub})^{y_{W^*}})$. Finally, the challenge ciphertext CT_{W^*} is send to $\mathcal{A}_1$.

Phase 2: The same as **Phase 1**.

Guess: $\mathcal{A}_1$ outputs a guess $b' \in \{0, 1\}$, and then $\mathcal{A}_2$ concludes its own game by outputting b'. Thus, we have $Adv_{\mathcal{A}_2}^{VBFIBE}(1^\kappa) = Adv_{\mathcal{A}_1}^{TFDAC-MACS}(1^\kappa)$. Based on the Lemma. 1, we know that the **Type-I** adversary has non-negligible advantage against the proposed scheme, which completes the proof of this theorem. ∎

Theorem 2: Suppose that the DBDH assumption holds in G. Then, there is no PPT **Type-I** adversary who can break the security of our TFDAC-MACS with non-negligible advantage.

This theorem follows directly from Lemma 1 and Theorem 1.

For **Type-II** Security: Here we prove that the proposed TFDAC-MACS is provable security against the static **Type-II** adversary under the decisional n-BDHE assumption.

Theorem 3: Suppose that the decisional n-BDHE assumption holds in G. Then, there is no PPT **Type-II** adversary who can break the security of our proposed scheme with non-negligible advantage.

Proof: Suppose that there exists a **Type-II** adversary $\mathcal{A}$, which breaks the proposed scheme with $Adv_{\mathcal{A}}(1^\kappa)^{TFDAC-MACS} \geq \epsilon$. We can build a simulator $\mathcal{B}$ that has advantage ϵ in solving the decision n-BDHE problem in G.

Assume that $\mathcal{A}$ chooses $W^* = \{W_1^*, \cdots, W_m^*\}$ as the challenge access policy. Let $I_{W^*}^U = \{i_1, \cdots, i_m\}$ denote the index set of attributes specified in W^*. The number of attributes that managed by AA_{aid} in W^* is denoted by $n_{W^*}^{aid}$. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

Setup. $\mathcal{B}$ takes a random decisional n-BDHE challenge $(g, h, \overrightarrow{y}_{g,\alpha,n}, Z)$ as input, and generates the public parameters. $\mathcal{B}$ randomly chooses $k^* \in [m]$, $a \in Z_p^*$ and $x_i, y_{i,j} \in Z_p^*$ for $i \in [n], j \in [n_i]$. For $k \in [m]$, $\mathcal{B}$ randomly chooses $x_{i_k}^*, y_{i_k}^* \in Z_p^*$. Assume that the attribute $u_{i_{k^*}}$ is managed by AA_{aid^*}.

For each AA_{aid}, $\mathcal{B}$ computes the public key APK_{aid} as follows:

1) For $aid \in I_{W^*}^A - \{aid^*\}$, $\mathcal{B}$ computes $APK_{aid} = e(g,g)^{x_{aid}^*}$.

2) For aid^*, $\mathcal{B}$ computes $APK_{aid^*} = e(g,g)^{x_{aid}^*} \cdot e(g,g)^{\alpha^{n+1}}$.

3) For $aid \notin I_{W^*}^A$, $\mathcal{B}$ computes $APK_{aid} = e(g,g)^{x_{aid}}$.

For $i_k \in I_{W^*}^U - \{i_{k^*}\}$, suppose $u_{i_k} \in U_{aid}$, $\mathcal{B}$ computes the public key $UPK_{i_k,j}$ for attribute value $v_{i_k,j}$ $(j \in [n_{i_k}])$ as follows:

1) If $v_{i_k,j} = W_{i_k}$ and $aid \neq aid^*$, $\mathcal{B}$ computes $UPK_{i_k,j} = g^{y_{i_k}^*} g_{n+1-i_k}^{-n_{W^*}^{aid^*}}$. If $v_{i_k,j} = W_{i_k}$ and $aid = aid^*$, $\mathcal{B}$ computes $UPK_{i_k,j} = g^{y_{i_k}^*} \cdot \prod_{t \in I_{W^*}^U - \{i_{k^*}\}} g_{n+1-t}$.

2) If $W_{i_k} \neq v_{i_k,j}$ and $aid \neq aid^*$, $\mathcal{B}$ computes $UPK_{i_k,j} = g^{y_{i_k,j}}$. If $W_{i_k} \neq v_{i_k,j}$ and $aid = aid^*$, $\mathcal{B}$ computes $UPK_{i_k,j} = g^{y_{i_k,j}} \cdot \prod_{t \in I_{W^*}^U - \{i_{k^*}\}} g_{n+1-t}$.

For i_{k^*}, $\mathcal{B}$ computes the public key $UPK_{i_{k^*},j}$ for attribute value $v_{i_{k^*},j}$ $(j \in [n_{i_{k^*}}])$ $(j \in [n_{i_k}])$ as follows:

1) If $v_{i_{k^*},j} = W_{i_{k^*}}$, then $UPK_{i_{k^*},j} = g^{y_{i_{k^*}}^*} \cdot \prod_{t \in I_{W^*}^U - \{i_{k^*}\}} g_{n+1-t}$.

2) If $v_{i_{k^*},j} \neq W_{i_{k^*}}$, then $UPK_{i_{k^*},j} = g^{y_{i_{k^*},j}} \cdot \prod_{t \in I_{W^*}^U - \{i_{k^*}\}} g_{n+1-t}$.

For $i_k \notin I_{W^*}^U$, suppose $u_{i_k} \in U_{aid}$, $\mathcal{B}$ computes the public key $UPK_{i_k,j}$ for attribute value $v_{i_k,j}$ as follows:

1) If $aid \neq aid^*$, $\mathcal{B}$ computes $UPK_{i_k,j} = g^{y_{i_k,j}}$.

2) If $aid = aid^*$, $\mathcal{B}$ computes $UPK_{i_k,j} = g^{y_{i_k,j}} \cdot \prod_{t \in I_{W^*}^U - \{i_{k^*}\}} g_{n+1-t}$.

Then, $\mathcal{B}$ sends $(g^a, \bigcup APK_{aid}, \bigcup UPK_{i_l,j})$ to $\mathcal{A}$, where $i_l \in [n], j \in [n_{i_l}]$.

Phase 1. $\mathcal{A}$ makes the following two queries for obtaining the attribute secret keys and authorization key. During this phase, $\mathcal{B}$ simulate the random oracles H for $\mathcal{A}$'s queries by maintaining a table L.

- **Auth query**: When there is a query on the authorization key for a global unique identifier uid, $\mathcal{B}$ needs to look for whether an item containing uid has been in L. If there is no such item, $\mathcal{B}$ randomly chooses $i_{k'} \in [n], z \in Z_p^*$, and adds the item $(uid, i_{k'}, z, H(uid) = g_{i_{k'}} g^z)$ into L and return $(g_{i_{k'}} g^z)^a$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ extract the last

component of this item, denoted by $H(uid)$, and returns $H(uid)^a$ to $\mathcal{A}$.

- **AAKeyGen query**: Suppose $\mathcal{A}$ submits an attribute list L_{uid} to $\mathcal{B}$ for a query on attribute secret key, where uid is a user's global unique identifier and $L_{uid} \not\models W^*$. For L_{uid}, there must exist $i_k \in I^U_{W^*}$ such that $L_{i_k} \neq W_{i_k}$. Without loss of generality, we assume that $L_{i_{\widehat{k}}} = v_{i_{\widehat{k}},j}$ and $W_{i_{\widehat{k}}} \neq v_{i_{\widehat{k}},j}$.
$\mathcal{B}$ first retrieves the item which contains uid in L. If such an item $(uid, i_{k'}, z, H(uid) = g_{i_{k'}} g^z)$ exists, then $\mathcal{B}$ computes the attribute secret keys associated with L_{uid}.
For $u_{i_{\widehat{k}}}$, suppose $u_{i_{\widehat{k}}} \in U_{aid}$, $\mathcal{B}$ computes the attribute secret key of $v_{i_{\widehat{k}},j}$ as follows:
1) If $i_{\widehat{k}} \neq i_{k^*}$ and $aid \in I^A_{W^*} - \{aid^*\}$, $\mathcal{B}$ computes $SK_{v_{i_{\widehat{k}},j}} = g^{x^*_{aid}} (g_{i_{k'}} g^z)^{-y_{i_{\widehat{k}},j}}$.
2) If $i_{\widehat{k}} \neq i_{k^*}$ and $aid = aid^*$, $\mathcal{B}$ computes $SK_{v_{i_{\widehat{k}},j}} = g^{x^*_{aid}} (g_{i_{k'}})^{-y_{i_{\widehat{k}},j}} \cdot (\prod_{i_k \in I^U_{W^*} - \{i_{k^*}, i_{k'}\}} g^{-1}_{n+1-i_k+i_{k'}})(UPK_{i_{\widehat{k}},j})^{-z}$.
3) If $i_{\widehat{k}} = i_{k^*}$, $\mathcal{B}$ computes $SK_{v_{i_{\widehat{k}},j}} = g^{x^*_{aid}} (g_{i_{k'}})^{-y_{i_{\widehat{k}},j}} (\prod_{i_k \in I^U_{W^*} - \{i_{k^*}, i_{k'}\}} g^{-1}_{n+1-i_k+i_{k'}}) \cdot (UPK_{i_{k^*},j})^{-z}$.
For $i_k \in I^U_{W^*}$ and $k \neq \widehat{k}$, suppose $u_{i_k} \in U_{aid'}$, $\mathcal{B}$ computes the attribute secret key of $v_{i_k,j} = W_{i_k} \in L_{uid}$ according to the following cases:
1) If $i_k \neq i_{k^*}$ and $aid' \neq aid^*$, $\mathcal{B}$ computes $SK_{v_{i_k,j}} = g^{x^*_{i_k}} (g_{i_{k'}})^{-y^*_{i_k}} g^{n^{aid}_{W^*}}_{n+1-i_k+i_{k'}} (UPK_{i_k,j})^{-z}$.
2) If $aid' = aid^*$, $\mathcal{B}$ computes $SK_{v_{i_k,j}} = g^{x^*_{aid}} (g_{i_{k'}})^{-y^*_{i_k}} (\prod_{i_k \in I^U_{W^*} - \{i_{k^*}, i_{k'}\}} g^{-1}_{n+1-i_k+i_{k'}}) \cdot (UPK_{i_k,j})^{-z}$.
For $i_k \notin I^U_{W^*}$, suppose $u_{i_k} \in U_{aid''}$, $\mathcal{B}$ computes the attribute secret key of $v_{i_k,j} \in L_{uid}$ as follows:
1) If $aid \in I^A_{W^*} - \{aid^*\}$, $\mathcal{B}$ computes $SK_{v_{i_k,j}} = g^{x^*_{aid}} (g_{i_{k'}} g^z)^{-y_{i_k,j}}$.
2) If $aid = aid^*$, $\mathcal{B}$ computes $SK_{v_{i_k,j}} = g^{x^*_{aid}} (g_{i_{k'}})^{-y_{i_k,j}} (\prod_{i_k \in I^U_{W^*} - \{i_{k^*}, i_{k'}\}} g^{-1}_{n+1-i_k+i_{k'}}) \cdot (UPK_{i_k,j})^{-z}$.
3) If $aid \notin I^A_{W^*}$, $\mathcal{B}$ computes $SK_{v_{i_k,j}} = g^{x_{aid}} (g_{i_{k'}} g^z)^{-y_{i_k,j}}$.
If there is no an item containing uid in L, $\mathcal{B}$ generates an item $(uid, i_{\widehat{k}}, z, H(uid) = g_{i_{\widehat{k}}} g^z)$ in L and then computes the attribute secret keys with respect to L_{uid} as above.
Finally, $\mathcal{B}$ sends $SK_{L_{uid}} = \{SK_{v_{i_k,j}} | v_{i_k} \in L_{uid}\}$ to $\mathcal{A}$.

Challenge. $\mathcal{B}$ sets $x^* = \sum_{aid \in I^A_{W^*}} (n^{aid}_{W^*} x_{aid})$ and $y^* = y^*_{i_1} + \cdots + y^*_{i_m}$, and aggregates the public attribute keys as:

$$APK_{W^*} = APK^{n^{aid^*}_{W^*}}_{aid^*} \prod\nolimits_{aid \in I^A_{W^*} - \{aid^*\}} APK^{n^{aid}_{W^*}}_{aid}$$

$$= e(g,g)^{x^* + n^{aid^*}_{W^*} \alpha^{q+1}},$$

$$UPK_{W^*} = UPK_{i_{k^*},j} \prod\nolimits_{i_k \in I_{W^*} - \{i_{k^*}\}} UPK_{i_k,j} = g^{y^*}.$$

After receiving two messages M_0 and M_1 of equal length submitted by $\mathcal{A}$, $\mathcal{B}$ picks a random bit $b \in \{0,1\}$ and computes the challenge ciphertext $CT_{W^*} = (W^*, C_1 = M_b Z^{n^{aid^*}_{W^*}} e(g,h)^{x^*}, C_2 = h, C_3 = h^{y^*} UPK_{W^*}{}^a)$.

Phase 2. The same as **Phase 1**.

Guess. $\mathcal{A}$ outputs a guess bit b' of b. If $b' = b$, $\mathcal{B}$ outputs 1 in the decisional n-BDHE game to guess that $Z = e(g_{n+1}, h)$. Otherwise, it outputs 0 to indicate that Z is a random element in G_T. Thus, if $Z = e(g_{n+1}, h)$, then CT_{W^*} is a valid ciphertext and we have $\Pr[\mathcal{B}(g, h, \overrightarrow{y}_{g,\alpha,n}, e(g_{n+1}, h)) = 1] = \frac{1}{2} + Adv^{TFDAC-MACS}_{\mathcal{A}}(1^\kappa) \geq \frac{1}{2} + \epsilon$.

If Z is a random element in G_T, the message M_b is completely hidden from $\mathcal{A}$ and we have $\Pr[\mathcal{B}(g, h, \overrightarrow{y}_{g,\alpha,n}, Z) = 1] = \frac{1}{2}$.

Therefore, $\mathcal{B}$ has advantage at least ϵ in solving the decisional n-BDHE problem in G.

By the n-BDHE assumption, we know that the proposed scheme is secure against the **Type-II** adversary. Thus, we complete the proof of this theorem. ■

Data Confidentiality: The above theorems only prove that the data confidentiality of our proposed TFDAC-MACS can be guaranteed against the unauthorized users or authorized users with insufficient attributes. For the CSP, it cannot properly decrypt any ciphertext since the decryption algorithm involves the attribute secret keys and authorization key. Although the CSP executes the process of ciphertext update, the CSP could not have the ability to decrypt any ciphertext. Because the CSP only uses the ciphertext update components to re-encrypt the involved ciphertexts with blindness of master secret keys and new authorization secret key. Therefore, data confidentiality against the CSP is guaranteed.

Collusion Resistance: The main challenge in designing an ABE scheme is to prevent against attacks from colluding users. In the proposed scheme, to decrypt a ciphertext, the colluding users should compute the value of $\prod_{v_{aid_i,j} \in W} e(g,g)^{y_{aid_i,j} s}$. To obtain this value, the colluding users need to aggregate the attribute secret keys that corresponding to access policy W, and then pair the C_2 from the ciphertext. But, the attribute secret keys of a user are related to the hash value of his/her global unique identity in the system. Therefore, the attribute secret keys of different users cannot be aggregated together for decryption. This means our proposed TFDAC-MACS can resist the collusion attack.

Forward Security: When a new user joins into the system, the attribute secret keys and authorization key of this user are all corresponding to the updated public attribute keys and new authorization secret key, respectively. So he/she can still decrypt previous ciphertexts, only if his/her attributes satisfy the access policies. Therefore, the forward security of the data is guaranteed in our proposed TFDAC-MACS.

Backward Security: If a user drops an attribute from his/her attribute set, this user cannot decrypt the previous ciphertexts, unless the remaining attributes satisfy the access policies. It consists of two reasons. One is that any involved AA_{aid} does not generate the corresponding attribute update key for this user, and the other is the CSP re-encrypts these ciphertexts referred to this revoked attribute value. Due to

TABLE 1. Security comparison between our scheme and other schemes.

Scheme	Against User	Against CSP	Collusion Resistance	Backward Security	Forward Security	Secure Channel in Revocation
DAC-MACS [2]	×	✓	×	×	✓	✓
Chen et al. [36]	×	✓	✓	×	✓	✓
MAACS [37]	✓	✓	✓	✓	✓	✓
NEDAC-MACS [35]	✓	✓	✓	✓	✓	×
TFDAC-MACS	✓	✓	✓	✓	✓	×

TABLE 2. Notations used in performance comparison.

Notation	Descriptions
p	the bit size of an element in Z_p, G and G_T with prime order p.
N	the bit size of an element in the group Z_N with composite order $N = p_1p_2p_3$.
n_A	the number of involved AAs in the system
n_{uid}	the number of attributes values held by DC_{uid}.
l	the number of rows of the access structure M in LSSS.
F_{uid}	the index set of the AAs to which the DC_{uid}'s attributes are related.
t_e	computation cost of one exponentiation operation
t_p	computation cost of one pairing operation

TABLE 3. Performance comparisons of MA-CP-ABE schemes.

Scheme	Access Policy	Parameter Size		Encryption Cost	Decryption Cost		Revocation	
		Secret Keys	Ciphertext		CSP	User	Attribute-level	User-level
DAC-MACS [2]	LSSS	$(3n_A + n_{uid} + 1)p$	$(3l + 3)p$	$(3l + 3)t_e$	$(2n_A + 3n_{uid})t_e + (n_A + n_{uid})t_e$	t_e	✓	×
Chen et al. [36]	AND_m	$n_{uid}p$	$3p$	$3t_e$	–	$2t_p$	✓	×
MAACS [37]	LSSS	$(3 + F_{uid} + n_{uid})N$	$(2l + 1)N + 1p_1$	$(3l + 2)t_e$	$n_{uid}t_e + (2n_{uid} + 1)t_p$	$(n_{uid} + 1)t_e$	✓	×
NEDAC-MACS [35]	LSSS	$(2n_A + 2n_{uid} + 1)p$	$(3l + 3)p$	$(3l + 3)t_e$	$(2n_A + 3n_{uid})t_e + (n_A + n_{uid})t_e$	t_e	✓	×
TFDAC-MACS	AND_m	$(n_{uid} + 1)p$	$3p$	$3t_e$	–	$3t_p$	✓	✓

the user's blindness of $x'_{aid_i,j}$ and r, this user cannot update the secret key of this attribute value and reverse the new ciphertext back to previous non-revoked state. Therefore, a user that one of his/her attributes is revoked and the rest attribute values are insufficient for the access policy cannot recover the outsourced data.

When a user is revoked by the data owner, the data owner does not generate the authorization update key for this revoked user. Furthermore, the CSP also re-encrypts the data owner's ciphertexts that bring these ciphertexts into correspondence with the new authorization key. These two points cause the revoked user cannot update his/her authorization key and reverse the new ciphertext back to previous non-revoked state, respectively. Thus, the revoked user cannot recover the outsourced data.

In a word, the proposed TFDAC-MACS can guarantee the backward security.

Table 1 details the comprehensive security comparisons between our proposed TFDAC-MACS and some existing CP-ABE schemes in multi-authority cloud storage systems. It is noted that our proposed TFDAC-MACS and NEDAC-MACS [35] does not need to a secure channel in revocation phase.

B. PERFORMANCE COMPARISON

To evaluate the efficiency, we carry out the performance comparisons among the five schemes in Table 1. Some notations that are used in the performance comparisons are briefly shown in Table 2.

Table 3 details the comparison results from the access policy, the parameter size, the encryption cost, the decryption cost and the type of revocation mechanisms. Our proposed TFDAC-MACS and Chen et al's scheme [36] only support the AND_m access policy, while DAC-MACS [2], MAACS [37] and NEDAC-MACS [35] all support the LSSS-realized access policy. It may bring some performance advantages for our proposed TFDAC-MACS and Chen et al's scheme [36], due to the the simplicity of AND_m access policy. Except for the MAACS [37], the rest schemes are all constructed in prime order bilinear group. As shown in Table 3, only our TFDAC-MACS achieves both the attribute-level revocation and user-level revocation. It also can be figured out that the storage cost of each user in our TFDAC-MACS is higher than that in Chen at al.'s scheme [36]. The reason is that each user still needs to store the authorization key in our TFDAC-MACS except for the attribute secret keys. We know that the ciphertext size implies the communication overhead in the system. The size of ciphertext in our TFDAC-MACS and Chen at al.'s scheme [36] are constant, while that in DAC-MACS [2], MAACS [37] and NEDAC-MACS [35] are linear to the scale of access policy. Therefore, our TFDAC-MACS and Chen at al.'s scheme incurs less communication overhead than the other three schemes. Since the pairing operation and exponential operation consume more

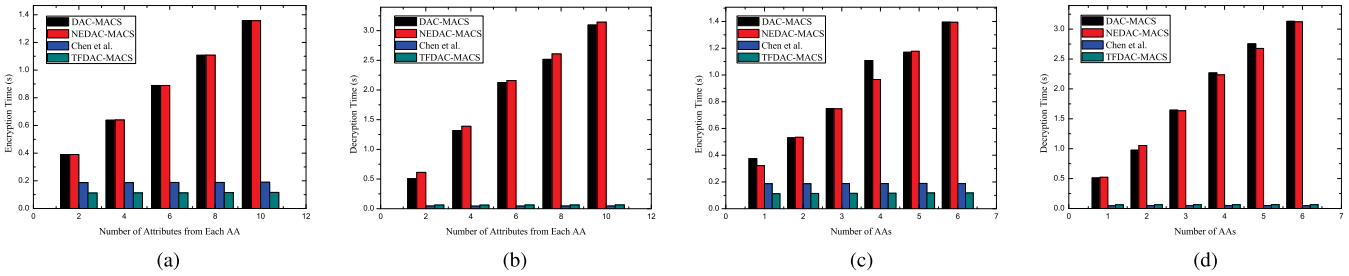


FIGURE 2. Comparisons of encryption time and decryption time. (a) Encryption. (b) Decryption. (c) Encryption. (d) Decryption.

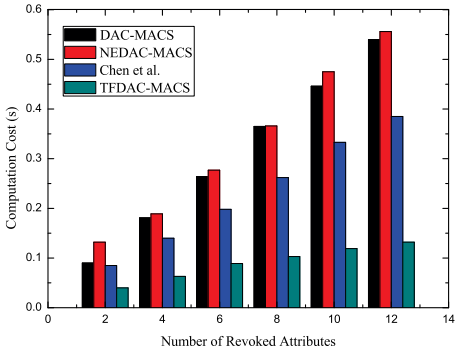


FIGURE 3. Comparison of attribute revocation.

computation cost than other operations, we only consider the computation cost of these two operations in the process of encryption and decryption. The encryption cost of data owner in our TFDAC-MACS is the same as that of Chen er al.'s scheme [36], which is lower than that of DAC-MACS [2], MAACS [37] and NEDAC-MACS [35]. Indeed, our TFDAC-MACS needs less multiplication operation than Chen et al.'s scheme [36] in computing the ciphertext component C_1. The decryption cost of users in DAC-MACS [2], MAACS [37] and NEDAC-MACS [35] are greatly alleviated by partitioning the computationally jobs of decryption and outsourcing the complicated bilinear pairing operations to the CSP. In our TFDAC-MACS, the job of decryption is only done by each user, where the decryption cost of each user is $3t_p$.

For precisely evaluating the computation costs, we implement our TFDAC-MACS and other three schemes with the Miracl library. We use an HP workstation equipped with 3.6GHz Intel Core CPU and 8GB Memory to simulate the CSP and AAs. The platform for data owner and user is a laptop with 2.53GHz Intel Core CPU and 2GB Memory. All the simulation platforms runs the Windows 7 Professional 64-bit operating system. In our simulation experiments, we set p is a 160-bit prime and the number of users is 50. The total number of universe attributes in the system is set as 60.

Fig. 2 describes the computation costs of encryption and decryption. They are somehow similar to the theoretical ones. In Fig. 2(a) and Fig. 2(b), the number of AAs is fixed to 6. And we set 10 as the number of involved users' attributes from each AA in Fig. 2(c) and Fig. 2(d). As shown in Fig. 2(a) and Fig. 2(c), the encryption costs of DAC-MACS [2] and NEDAC-MACS [35] are linear to the scale of access policy in ciphertext, while that of Chen et al.'s scheme [36] and our TFDAC-MACS is nearly a constant value. We can see that the proposed TFDAC-MACS has the minimum overhead on encryption. In Fig. 2(b) and Fig. 2(d), the decryption time for DAC-MACS [2] and NEDAC-MACS [35] includes the computation time of CSP. Chen et al's scheme [36] is superior to that of the remaining schemes in decryption cost, and Our TFDAC-MACS incurs less computation time of decryption than DAC-MACS [2] and NEDAC-MACS [35]. But, users in DAC-MACS [2] and NEDAC-MACS [35] only spend very small computation cost on decryption since the most computation overhead of decryption is done by the CSP. In our simulation experiments, the number of users who hold the revoked attribute is set as 2. Fig. 3 shows the comparison of computation cost in attribute-level revocation. In summary, our TFDAC-MACS is efficient through the comparison results. In our simulation experiments, the computation cost of user-level revocation for our TFDAC-MACS is 0.616s when revoking one user. Maybe this result is a little expensive. Because the data owner in user revocation needs to compute a large number of exponentiation operations.

VI. CONCLUSION

In this paper, we propose a new data access control scheme for multi-authority cloud storage systems. The proposed scheme provides two-factor protection mechanism to enhances the confidentiality of outsourced data. If a user want to recover the outsourced data, this user is required to hold sufficient attribute secret keys with respect to the access policy and authorization key with regard to the outsourced data. In our proposed scheme, both the size of ciphertext and the number of pairing operations in decryption are constant, which reduce the communication overhead and computation cost of the system. In addition, the proposed scheme provides the user-level revocation for data owner in attribute-based data access control systems. Extensive security analysis, performance comparisons and experimental results indicate that the proposed scheme is suitable to data access control for multi-authority cloud storage systems.

REFERENCES

[1] M. Armbrust *et al.*, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] K. Yang, X. Jia, K. Ren, B. Zhang, and R. Xie, "DAC-MACS: Effective data access control for multiauthority cloud storage systems," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1790–1801, Nov. 2013.

[3] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, "New publicly verifiable databases with efficient updates," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 5, pp. 546–556, Sep. 2015.

[4] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Comput.*, vol. 16, no. 1, pp. 69–73, Jan./Feb. 2012.

[5] S. Kavitha and S. Subashini, "A survey on security issues in service delivery models of cloud computing," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 1–11, Jan. 2011.

[6] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *Proc. 1st Workshop Real-Life Cryptograph. Protocols Standardization (RLCPS)*, vol. 6054. 2010, pp. 136–149.

[7] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms for secure outsourcing of modular exponentiations," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2386–2396, Sep. 2014.

[8] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. CRYPTO*, vol. 2139. 2001, pp. 213–229.

[9] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 3494. Heidelberg, Germany: Springer-Verlag, 2005, pp. 457–473.

[10] 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. (CCS)*, Oct./Nov. 2006, pp. 89–98.

[11] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 7, pp. 1214–1221, Nov. 2011.

[12] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 8, pp. 1343–1354, Aug. 2013.

[13] K. Yang, X. Jia, and K. Ren, "Attribute-based fine-grained access control with efficient revocation in cloud storage systems," in *Proc. 8th ACM SIGSAC Symp. Inf., Comput. Commun. Secur. (ASIACCS)*, New York, NY, USA, 2013, pp. 523–528.

[14] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proc. 5th ACM Symp. Inf., Comput. Commun. Secur. (ASIACCS)*, New York, NY, USA, 2010, pp. 261–270.

[15] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 321–334.

[16] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, Oct. 2007, pp. 456–465.

[17] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, "CP-ABE with constant-size keys for lightweight devices," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 5, pp. 763–771, May 2014.

[18] J. Li *et al.*, "Fine-grained data access control systems with user accountability in cloud computing," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Nov./Dec. 2010, pp. 89–96.

[19] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2007, pp. 195–203.

[20] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. 14th Int. Conf. Pract. Theory Public Key Cryptogr. (PKC)*, vol. 6571. 2011, pp. 321–334.

[21] A. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," in *Advances in Cryptology*, vol. 7417. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 180–198.

[22] M. Chase, "Multi-authority attribute based encryption," in *Proc. 4th IACR Theory Cryptogr. Conf. (TCC)*, vol. 4392. Feb. 2007, pp. 515–534.

[23] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," in *Proc. 9th Int. Conf. Cryptol. India (INDOCRYPT)*, vol. 5365. 2008, pp. 426–436.

[24] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 6632. Heidelberg, Germany: Springer, 2011.

[25] 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. 6th ACM Symp. Inf., Comput. Commun. Secur. (ASIACCS)*, New York, NY, USA, 2011, pp. 386–390.

[26] 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. 16th Eur. Conf. Res. Comput. Secur. (ESORICS)*, vol. 6879. 2011, pp. 278–297.

[27] Y. Rouselakis and B. Waters, "Efficient statically-secure large-universe multi-authority attribute-based encryption," in *Proc. 19th Int. Conf. Financial Cryptogr. Data Secur. (FC)*, vol. 8975. 2015, pp. 315–332.

[28] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *Proc. 5th Int. Conf. Inf. Secur. Pract. Exper.*, vol. 5451. Apr. 2009, pp. 13–23.

[29] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *Public Key Cryptography* (Lecture Notes in Computer Science), vol. 6056. Heidelberg, Germany: Springer-Verlag, 2010, pp. 19–34.

[30] C. Chen, Z. Zhang, and D. Feng, "Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost," in *Proc. 5th Int. Conf. Provable Secur. (ProvSec)*, vol. 6980. 2011, pp. 84–101.

[31] Y. S. Rao and R. Dutta, "Recipient anonymous ciphertext-policy attribute based encryption," in *Proc. 9th Int. Conf. Inf. Syst. Secur. (ICISS)*, vol. 8303. 2013, pp. 329–344.

[32] Y. Zhang, X. Chen, J. Li, H. Li, and F. Li, "Attribute-based data sharing with flexible and direct revocation in cloud computing," *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 11, pp. 4028–4049, 2014.

[33] N. Attrapadung and H. Imai, "Conjunctive broadcast and attribute-based encryption," in *Proc. 3rd Int. Conf. Palo Alto Pairing-Based Cryptogr. (Pairing)*, vol. 5671. 2009, pp. 248–265.

[34] K. Yang and X. Jia, "Attributed-based access control for multi-authority systems in cloud storage," in *Proc. IEEE 32nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2012, pp. 536–545.

[35] X. Wu, R. Jiang, and B. Bhargava, "On the security of data access control for multiauthority cloud storage systems," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2015.2441698.

[36] C. Yanli, S. Lingling, and Y. Geng, "Attribute-based access control for multi-authority systems with constant size ciphertext in cloud computing," *China Commun.*, vol. 13, no. 2, pp. 146–162, Feb. 2016.

[37] Q. Li, J. Ma, R. Li, X. Liu, J. Xiong, and D. Chen, "Secure, efficient and revocable multi-authority access control system in cloud storage," *Comput. Secur.*, vol. 59, pp. 45–59, Jun. 2016.

[38] J. K. Liu, K. Liang, W. Susilo, J. Liu, and Y. Xiang, "Two-factor data security protection mechanism for cloud storage system," *IEEE Trans. Comput.*, vol. 65, no. 6, pp. 1992–2004, Jun. 2016.

[39] D. Boneh, X. Boyen, and E. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Advances in Cryptology—EUROCRYPT*, vol. 3494. Berlin, Germany: Springer-Verlag, 2005, pp. 440–456.

[40] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Proc. 6th Int. Conf. Appl. Cryptogr. Netw. Secur.*, vol. 5037. Jun. 2008, pp. 111–129.

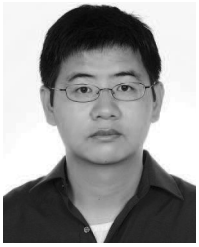
[41] D. Boneh and X. Boyen, "Efficient selective-ID secure identity-nased encryption without random oracles," in *Advances in Cryptology—EUROCRYPT*, vol. 3027. Berlin, Germany: Springer-Verlag, 2004, pp. 223–238.

XIAOYU LI received the B.S. degree in information and computing science from Hubei Normal University, in 2009, and the M.S. degree in computer science from Guangxi Normal University, in 2012. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, South China University of Technology, China. His current research interests include applied cryptography, cloud security, and privacy protection.

SHAOHUA TANG received the B.S. and M.S. degrees in applied mathematics and the Ph.D. degree in communication and information system from the South China University of Technology, in 1991, 1994, and 1998, respectively. He has been a Full Professor with the School of Computer Science and Engineering, South China University of Technology, since 2004. His current research interests include information security, networking, and information processing.

LINGLING XU received the B.S. and M.S. degrees in mathematics from Shandong University, China, in 2005 and 2008, respectively, and the Ph.D. degree in communication and information system from Sun Yat-sen University in 2011. She is currently an Assistant Professor with the School of Computer Science and Engineering, South China University of Technology. Her current research interests include cryptography and cloud computing.

HUAQUN WANG received the B.S. degree in mathematics education from Shandong Normal University, the M.S. degree in applied mathematics from East China Normal University, China, in 1997 and 2000, respectively, and the Ph.D. degree in information security from the Nanjing University of Posts and Telecommunications, China, in 2006, where he is currently a Professor. His research interests include applied cryptography, network security, and cloud computing security.



JIE CHEN received the B.S. degree in mathematics from Soochow University, China, in 2008, and the Ph.D. degree in mathematics from Nanyang Technological University, Singapore, in 2012. He was a Research Associate with Nanyang Technological University from 2012 to 2013. He is currently a Professor with East China Normal University, China, and a Researcher with the Ecole Normale Supérieure de Lyon, France. His research interests include public-key cryptography and information security.