IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 1, 2019, accepted September 10, 2019,
date of publication September 19, 2019, date of current version October 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2942414

Security and Efficiency Enhanced Revocable Access Control for Fog-Based Smart Grid System

MI WEN[1], (Member, IEEE), SHAN CHEN[1], RONGXING LU[2], (Senior Member, IEEE), BEIBEI LI[3], (Member, IEEE), AND SIJIA CHEN[1]

[1]College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 200090, China
[2]Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada
[3]College of Cybersecurity, Sichuan University, Chengdu 610065, China

Corresponding author: Shan Chen (shanchen@mail.shiep.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant No. 61872230 and No. 61572311.

ABSTRACT With the popularity of smart grids, plentiful of smart devices have been put into use, such as smart meters and power assets. Due to limited computation capabilities and storage spaces of these devices, the collected data need to be ''outsourced'' towards the data server for processing and storage. The data owners, therefore, lose direct control over these ''outsourced'' data, leading to significant security issues of the users' data. In this paper, aiming at solving this problem, we propose a multi-authority Ciphertext Policy Attribute-based Encryption (CP-ABE) scheme with revocation for the fog-based smart grid system. Specifically, in order to achieve attribute revocation without requiring users to be always online, we use the DH (Diffie-Hellman) tree to distribute the group key statelessly, which also solves the problem of collusion attack initiated by revoked user and valid user. To improve security of our proposed scheme, we remove the trusted key authority (KA) by using a secure two-party computation (2PC) protocol between the KA and the cloud service provider to generate user private key. To improve efficiency of our proposed scheme, we combine user and attribute revocation, and outsource complex calculations to fog nodes. Furthermore, our proposed scheme uses attribute group key and leaf private key together to protect user proxy key, which reduces the storage overhead of the system and improves the security. Both security analysis and experimental results demonstrate that our proposed scheme can balance the security objectives with the efficiency.

INDEX TERMS CP-ABE scheme, fog computing, revocation, smart grid.

I. INTRODUCTION

Over the past few years, cloud computing has provided abundant supports for smart grid. For instance, electricity data can be processed and analyzed in the cloud to regulate electricity prices based on electricity usage. In order to make the smart grid more intelligent, a large number of smart terminals are deployed in order to collect the grid status for the control center in time [1]. However, these smart terminals, such as smart meters (SMs), will generate a large amount of data, resulting in the explosive growth of power consumption data, thus the current cloud computing paradigm is not sufficient to meet the heterogeneous, low-latency, and intensive service requirements [2].

Fog computing is a new paradigm, which has great advantages in real-time processing of massive data because of its more distributed network architecture [3]. For this reason, fog-based smart grid system has been proposed to address the above challenges. With the support of fog nodes, massive data from different smart meters can be collected and processed at the edge of the network, and then transmitted to and stored in cloud. Since all transmissions from the terminal nodes to the remote cloud cost more than forwarding them to a nearby fog node, the fog node can be effectively utilized to provide services with low latency, mobility, and location awareness [4]. However, since the data are handed over to fog nodes for processing and stored in cloud, the data will

The associate editor coordinating the review of this manuscript and approving it for publication was Giacomo Verticale.

be out of control from data owners. The smart grid is an organization that serves the public. If the user's data is not well protected, sensitive information such as user power consumption, phone number, and home address will be leaked, which will have a negative impact on public safety. Thus, the security problem of power data has become important in smart grid [5]. Whereas, existing data access control schemes in cloud computing cannot be applied to the scenario of fog-based smart grid, because they cannot meet the requirements of efficient and convenient communication in a three-layer network.

To achieve secure and efficient data sharing for fog-based smart grid, we apply the method of CP-ABE (Ciphertext Policy Attribute-Based Encryption) [6]. Specifically, CP-ABE scheme enables the encryptor to define a subset of attributes required by the decryptor to access the plaintext data, and encrypt the data with the set of attributes. Therefore, each user with different attribute sets is allowed to decrypt different ciphertexts according to the access policy, and it effectively eliminates the dependence on data storage servers to prevent unauthorized users from accessing data.

However, there are several challenges in applying CP-ABE to data sharing systems. In CP-ABE, the key authority (KA) generates each user's private key through the KA's master private key and that user's related attribute set. In this way, the KA no longer needs to process and store public key certificates like the traditional public key infrastructure (PKI), which is a key advantage of CP-ABE. However, it must be guaranteed that the KA is fully trusted, prior to taking the advantages of CP-ABE. The KA can decrypt each ciphertext sent to a specific user by generating the user's private key. This may be a potential threat to the data security in data sharing systems. Besides, there are many attributes generated not by a same KA. In the smart grid, the user's attributes may be issued by different authorities [7]. For example, the access policy of a ciphertext is (''Graduates of Shanghai University of Electric Power'' and ''Engineer of State Grid''). In this way, only the users who graduated from Shanghai University of Electric Power and are now employed as engineers by State Grid can decrypt the message. Shanghai University of Electric Power is responsible for issuing attributes to students, while State Grid is responsible for issuing attributes to its employees.

Another challenge is the revocation problem [8]. Since smart terminals may join or leave the system at any time, and their attributes will also be changed frequently, so it is necessary to revoke or update each attribute to make the system safe. This problem is more challenging in CP-ABE because each attribute can be shared by multiple users, which means that revoking any user in a attribute group affects all users in that group. Re-encrypting all affected ciphertext will incur unbearable overhead on users, so the revocation is a bottleneck in CP-ABE scheme. On the other hand, due to the nature of wireless networks, mobile users may suffer from transient connections, which can not guarantee the continuous availability of the system. Therefore, when revocation occurs, it is impossible for users to update their private keys online at all times.

System overhead is also a problem in applying CP-ABE scheme [9], because CP-ABE scheme requires high computation overhead during performing encryption and decryption operations. On the client side, many smart terminal devices such as smart meters have significantly limited onboard memory, processors, lifetime, and available network bandwidth compared to desktop computers. Hence, any protocol that provides additional security should not impose heavy costs on end users.

In order to address the above challenges, this paper proposes a multi-authority CP-ABE scheme with revocation, which removes the fully trusted authority center and is lightweight on the user side. The main contributions of this paper are three-fold:

- First, based on the attribute revocation scheme in the cloud [10], we design a new multi-authority CP-ABE scheme with revocation for the fog-based smart grid system. Particularly, we propose for the first time to construct a DH tree to distribute attribute group keys without requiring users to be always online. By this manner, the problem, can not resist collusion attack, in [10] is solved.
- Second, in order to improve the efficiency, we combine two granular revocation schemes and outsource complex calculations to fog nodes without revealing user data. To improve the security, we remove the fully trusted authority center in our scheme by a secure two-party computation (2PC) protocol. Besides, we associate the leaf private key with the user proxy key, without the need for the fog node to check the version of the leaf private key for all of the user's attributes. This not only reduces the storage overhead of our scheme, but also improves the security of our scheme.
- Third, through the security analysis, our scheme can achieve data confidentiality, forward and backward security, and resist collusion attack. Experiments and performance results show that it only generates a small overhead on the user side.

The remainder of this paper is organized as follows: we discuss the related work in Section II, and introduce the system model and security requirements in Section III. Then, we describe some preliminaries in Section IV. In Section V, we present our scheme, followed by security analysis and performance analysis in Section VI and Section VII, respectively. Finally, we draw our conclusion in Section VIII.

II. RELATED WORK

To achieve secure and flexible fine-grained access control, Sahai and Waters first introduce the concept of ABE [11], in which data owners can share their private data with prospective users without knowing their exact public key or identity. And the ABE scheme is mainly divided into two categories: Key Policy Attribute-based Encryption (KP-ABE) [12] and Ciphertext Policy Attribute-based

Encryption (CP-ABE) [6]. In KP-ABE scheme, access structures are specified in the private key, and attributes are used to describe ciphertext. Whereas the roles of the ciphertext and the key are reversed in CP-ABE scheme, where ciphertext is encrypted using an access policy selected by the encryptor, but the key is simply created relative to user's attribute set. Between these two methods, CP-ABE is more suitable for data sharing systems because it gives access policy decisions to the data owner. Thus in this paper, we will only consider CP-ABE scheme.

A. KEY AUTHORITY

All of the previous work described above only considers the case where all attributes are monitored by a single privilege. However, as we mentioned in Section I, one might want to divide the control of various attributes into many different authorities. Chase [13] proposes a multi-authority ABE scheme, which transfers the generation of private keys to multiple authorities to complete together. Ramesh and Priya [14], Fan *et al.* [15] propose a multi-authority CP-ABE scheme, in which every attribute authority should register itself to the KA, the KA then assigns a unique global authority identity *AID* to each legitimate attribute authority. After that, the attribute authority uses its *AID* to generate the attribute key for users. However, these schemes are still not ideal because they require a fully trusted authority center. Chase and Chow [16] propose a distributed ABE scheme, which removes the fully trusted authority center in multi-authority systems. One disadvantage of this fully distributed approach is that all attribute authorities should communicate with other authorities in the system to generate user private keys. As a result, performance degradation is caused. Nirmalrani *et al.* [17] propose a CP-ABE scheme, which removes the fully trusted authority center in single-authority systems. Therefore, we believe that designing an effective and secure multi-authority ABE scheme without a fully trusted authority center is still a very important problem, which is one of the problems we will try to solve in this paper.

B. REVOCATION

Recently, several CP-ABE schemes that can deal with revocation problems have been proposed. There are two granular revocation mechanisms, fine-grained attribute revocation and coarse-grained user revocation. Attribute-level revocation means that when a user loses some attributes, he can still access data as long as the remaining attributes satisfy the access policy. User-level revocation means that when a user is revoked, he loses all access rights.

Boldyreva *et al.* [18] and Pirretti *et al.* [19] propose some attribute-revocable ABE schemes, they realize the revocation by adding an expiration time to each attribute. However, in these schemes, new users may be able to access the previous data encrypted before they join the system until the data is re-encrypted by using the newly updated attribute key periodically. On the other hand, the revoked user can still access the encrypted data until the next expiration time, even if he no longer holds the attribute. To realize real-time revocation, Zhang *et al.* [8] propose a attribute-revocable CP-ABE schemes in fog by sending valid key update messages to the unrevoked users. However, their scheme requires all users to be online all the time to update group keys, and the communication overhead will become the performance bottleneck of this large-scale system. Hur and Noh [10], [17] provide a simple way to revoke attribute by combining CP-ABE with re-encryption. In their scheme, all users with the same attributes belong to a group, and hold the group key published by the group. The ciphertext will be re-encrypted by the group key, and the group key will be stored in head message implicitly. Thus, users can obtain the latest version of group keys from head messages without always be online. However, their scheme can not resist the collusion attack initiated by the valid user and the revoked user in a group. Because each user of the same attribute group has the same group key.

Yu *et al.* [20] and Fan *et al.* [15] propose a user-revocable scheme in ABE-based data sharing system. In this scheme, user revocation is implemented using proxy re-encryption of the data server. In order to revoke the user, the KA not only generates a user private key for the user, but also generates a proxy key for the data server, and the data server re-encrypts the ciphertext with the proxy key. When user revocation occurs, the data server will delete the proxy key of the user, thus the revoked user cannot decrypt the ciphertext. However, the problem that KA must be fully trusted in these schemes cannot be avoided, because the KA manages the private key of all users and the proxy key of the data server.

C. OUTSOURCING

Teng *et al.* [9], [21] propose some ABE schemes with constant-size ciphertext to reduce the computation cost in encryption and decryption. In order to further reduce the computing cost of resource-constrained devices, some cryptographic operations with high computation overhead can be outsourced to data servers. The fog node is one of the best agents to perform outsourcing services for users, because it is the edge of the cloud, and closer to the end user. Thus it can be used to reduce the computation overhead required for resource-constrained devices. Some CP-ABE schemes with outsourcing have been proposed recently [8], [22], [23]. Zhang *et al.* [8] construct a CP-ABE scheme with encryption and decryption outsourcing. In the encryption phase of this scheme, the data owner first performs partial encryption independent of access policy, and then uses semi-trusted proxy to perform encryption related to access policy. In decryption phase, the user sends the proxy key to the semi-trusted proxy to perform most attribute-related decryption operations, leaving only a constant number of simple calculations for the user. Because of outsourcing, the demand for computing power of user terminals is low, so that more resource-limited smart terminals can have longer lifecycle.

TABLE 1. Symbols.

Symbol	Explanation
T	Access tree defined by data owners
U	User set in the system
S	Attribute set in the system
G_j	Collection of users who have the attribute λ_j
K_{λ_j}	Group key shared by unrevoked users in the attribute group G_j

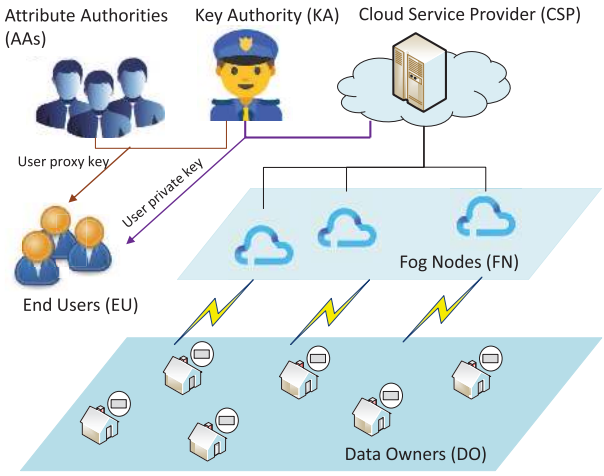


FIGURE 1. System model under consideration.

III. SYSTEM OVERVIEW

In this section, we formalize the system model and system security requirements. Some symbols used in this paper are described in Table 1.

A. SYSTEM MODEL

In smart grid, residential customers equipped with smart meters (SMs) generate a large amount of power consumption data that needs to be stored in the smart grid operation center, also known as cloud service provider (CSP) in our system model, they are data owners. After that, the user authorized by the data owner, such as the grid administrator, will access the stored power consumption data for analysis and adjustment. In this paper, we mainly focus on how to store and access power consumption data in a privacy-preserving manner. Specifically, we consider a network framework for smart grid as shown in Fig. 1, there exist the following six entities: Key Authority (KA), Attribute Authorities (AAs), Cloud Service Provider (CSP), Fog Nodes (FN), Data Owners (DO), and End Users (EU).

- *KA:* It is responsible for generating personalized user private keys with the CSP and proxy keys with AAs. It is also responsible for managing attribute groups, generating attribute group keys and head messages. In addition, the KA stores the proxy key for users, when user revocation occurs, it will remove the corresponding proxy key. It is assumed to be honest-but-curious. That is, it will honestly execute the assigned tasks in the system, but is curious about users' privacy data.
- *AAs:* Each AA is responsible for managing different attribute domains, which are a subset of the system attribute set. Also each AA is responsible for generating proxy keys with the KA for users within its administration domain. They are assumed to be honest-but-curious.
- *CSP:* It provides a series of services including data storage and access. In addition to generating personalized user private keys with the KA, it also stores ciphertext, head messages, and is responsible for updating ciphertext. It is assumed to be honest-but-curious.
- *FN:* Each fog node is connected to the cloud server and is a "bridge" between the user and the cloud server. It is responsible for providing outsourcing services to users: helping users to perform complex partial encryption and decryption operations. They are assumed to be honest-but-curious.
- *DO:* They have a large amount of data to be stored in the cloud. They are responsible for defining access policy and partially encrypting data before outsourcing it.
- *EU:* They want to access ciphertext stored in the cloud. When a valid user whose attributes satisfy the access policy in the ciphertext, he can decrypt the ciphertext.

B. SECURITY REQUIREMENTS

- *Data confidentiality:* unauthorized users who are not defined by the data owner as expected visitors should be prevented from accessing the data. In addition, the honest but curious KA, AAs, CSP, and FN should be prevented from unauthorized access to ciphertext.
- *Collusion resistance:* multiple unauthorized users may work together to decrypt a ciphertext that none of them can decrypt alone [24]. Our access system is required to be secured against such collusion attacks. In addition, collusion attacks initiated by revoked users and valid users in the same attribute group should also be defended.
- *Forward security and backward security:* backward security means that any user who obtains an attribute to satisfy the access policy should be prevented from accessing the previous data distributed before he holds the attribute. On the other hand, forward security means that unless the other valid attributes he holds still satisfy the access policy, any user who revokes the attributes should be prevented from accessing the data distributed after he revokes the attributes.

IV. PRELIMINARIES

In this section, we briefly recall four basic building blocks used in our proposed scheme, namely the access structure [6], the bilinear pairing [25], the tree-based group DH key [26], and the CP-ABE scheme without the fully trusted authority center [17].

A. ACCESS STRUCTURE

Let $\{P_1, P_2, \ldots, P_n\}$ be a set of parties. For $\forall B, C$: if $B \in A$ and $B \subseteq C$, then $C \in A$, the collection $A \subseteq 2^{\{P_1, P_2, \ldots, P_n\}}$ is monotone. An access structure A is a collection of nonempty subsets of $\{P_1, P_2, \ldots, P_n\}$. The sets in A are called the authorized sets, and the sets not in A are called the

unauthorized sets. In CP-ABE schemes, the role of the parties is determined by the attributes. In our scheme, private keys are bound to attribute sets and messages are encrypted through access trees.

1) ACCESS TREE *T*

Let T be a tree expressing an access policy. Each leaf node in the tree is expressed by an attribute and a threshold value $k_x = 1$. λ_x represents the attribute associated with the leaf node in the tree. Each non-leaf node in the tree is expressed by a threshold gate and its children. num_x denotes the number of children of node x, and k_x denotes the threshold value of node x, where $0 < k_x \leq num_x$. The children of a non-leaf node are enumerated from 1 to num_x. The function $index(x)$ returns the number of node x, where the index value is uniquely assigned to the node in the access policy.

2) SATISFYING AN ACCESS TREE

Assume T is an access tree with root R, T_x represents the subtree for T rooted in the node x. Therefore T can be represented as T_R. If attribute set S satisfies the access tree T_x, it is denoted as $T_x(S) = 1$. $T_x(S)$ is recursively calculated as follows: if x is a non-leaf node, we compute $T_z(S)$ for all children z of node x. $T_x(S)$ outputs 1 if and only if at least k_x children return 1. If x is a leaf node, then $T_x(S)$ outputs 1 if and only if $\lambda_x \in S$.

B. BILINEAR PAIRING

Let G and G_T be two cyclic groups with the same prime order p. A map $e : G \times G \rightarrow G_T$ is a bilinear map if it satisfies the following properties:

1) Bilinearity: for any $g_1, g_2 \in G$ and $a, b \in Z_p$, it has $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.

2) Nondegeneracy: $e(g_1, g_2) \neq 1$.

3) Computability: $e(g_1, g_2)$ can be efficiently computed.

C. TREE-BASED GROUP DH KEY

Let G be a cyclic group generated by g with the prime order p, and $\iota : G \rightarrow Z_p^*$ be an injection. Then, the tree-based group key is defined as follows:

1) Assume nodes X and Y respectively have their public-private key pairs $(pk = g^x \in G, sk = x \in Z_p^*)$ and $(pk = g^y \in G, sk = y \in Z_p^*)$, as shown in Fig. 2.

2) When nodes X and Y form a tree-based group $\{X, Y\}$, both of them can compute the group private key $sk = \iota(g^{xy}) \in Z_p^*$ with their private keys, and compute the group public key $pk = g^{\iota(g^{xy})} \in G$. For more details, please refer to [26].

D. THE CP-ABE SCHEME WITHOUT THE FULLY TRUSTED AUTHORITY CENTER

Since the KA is semi-trusted, it should be prevented from accessing the plaintext of the data to be shared. At the same time, it should still be able to issue keys to users. In order to achieve this somewhat contradictory requirement, the KA and the CSP use their own master private key to participate in the arithmetic 2PC protocol, and issue independent key components to users in the key generation phase. The 2PC protocol prevents them from knowing each other's master private keys so that they cannot generate the user's whole set of keys alone.The CP-ABE scheme without a fully trusted authority center consists of the following phases:

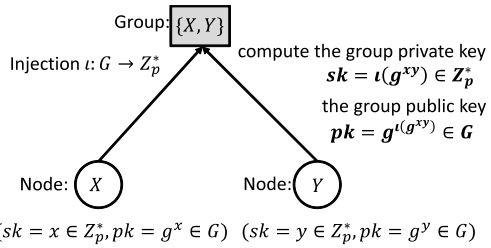


FIGURE 2. Tree-based group DH key technique.

1) SYSTEM INITIALIZATION

This phase is divided into three sub-phases.

i) The trusted initializer takes a security parameter as input. It outputs a public parameter $Param$.

ii) The KA takes nothing as input. It outputs its master public and private key pair PK_{KA}, MK_{KA}.

iii) The CSP takes nothing as input. It outputs its master public and private key pair PK_{CSP}, MK_{CSP}.

2) KEY GENERATION

This phase is also divided into three sub-phases.

i) The KA and the CSP engage in the arithmetic secure 2PC protocol to generate the user private key. The KA takes as input its master private key MK_{KA} and a user identity uid, and gets nothing as output. The CSP takes as input its master private key MK_{CSP} and a user identity uid, and then gets the key component $SK_{C,uid}$ as output.

ii) The KA takes as input the attribute set S_1 of a user uid, it outputs the key component $SK_{K,uid}$.

iii) The user uid gets $SK_{C,uid}$ form the CSP, and gets $SK_{K,uid}$ from the KA, then it gets the whole private key SK_{uid} by combining the two key components.

3) ENCRYPTION

The data owner takes the public parameter $Param$, a message M, and an access tree T as input. It outputs a ciphertext CT that only the user whose attribute set satisfies the access policy can decrypt.

4) DECRYPTION

The data user takes as input the public parameter $Param$, a ciphertext CT, and a whole private key SK_{uid}. If the user's attribute set S_1 satisfies the access tree T embedded in the CT, then it decrypts the ciphertext successfully and returns M.

V. PROPOSED CP-ABE SCHEME

As an improved multi-authority CP-ABE scheme with revocation mechanism and calculation outsourcing, our proposed

scheme mainly includes five phases, namely system initialization, key generation, encryption, decryption, and revocation mechanism. To eliminate trusted third-party issues, our scheme will generate user private keys in an interactive way. One of the sub-phases of key generation is to generate attribute group keys, the DH tree method will be adopted to solve the problem of collusion attack in the scheme [10]. Furthermore, our proposed scheme outsources complex operations to fog nodes, so the encryption and decryption phases are naturally divided into two parts: local operation and outsourcing operation.

A. SYSTEM INITIALIZATION

Without loss of generality, we assume there is a trust initializer who will initialize the whole system. Specifically, by taking a security parameter κ as input, the true initializer outputs a 5-tuple (p, G, G_T, g, e), where p is a κ-bit prime number, G and G_T are two groups with the same order p, $g \in G$ is a generator, and $e : G \times G \to G_T$ is a nondegenerated and efficiently computable bilinear map. It also selects $h \in G$ at random and a hash function $H : \{0, 1\}^* \to Z_p^*$. It outputs the public parameter as $Param = \{G, G_T, e, H, g, h\}$. The KA selects $\alpha \in Z_p^*$ at random, and outputs the master public and private key pair as $(PK_{KA} = g^{\alpha}, MK_{KA} = \alpha)$. The CSP selects $\beta \in Z_p^*$ at random, and outputs the master public and private key pair as $\left(PK_{CSP} = e\left(g, g\right)^{\beta}, MK_{CSP} = g^{\beta}\right)$.

B. KEY GENERATION

The KA and the CSP use their own master private keys to participate in a secure 2PC protocol to generate the user private key, and the 2PC protocol prevents them from knowing each other's master private keys so that they cannot generate the user private key alone. In addition, the KA is also responsible for generating the user proxy key and the attribute group key.

1) GENERATE USER PRIVATE KEY

The user private key generation protocol is as follows:

i) The KA selects a globally unique *uid* and $r \in Z_p^*$ for each legitimate user, which should be consistent for any further attribute additions to the user. Then, the KA and the CSP participate in a secure 2PC protocol where the KA's private input is (α, r) and the CSP's private input is β. The private output of the secure 2PC protocol is $x = (\beta + r)\,\alpha \ mod \ p$, which is returned to the CSP. This can be done via a general secure 2PC protocol for a simple arithmetic computation [16], [27]. Alternatively, we can do this more efficiently using the construction in [28].

ii) The CSP randomly selects $\tau \in Z_p^*$ and computes $A = g^{\frac{x}{\tau}} = g^{\frac{(\beta+r)\alpha}{\tau}}$. Then it sends A to the KA.

iii) The KA computes $B = A^{\frac{1}{\alpha^2}} = g^{\frac{\beta+r}{\tau\alpha}}$, then sends B to the CSP.

iv) The CSP outputs user private key as $SK_{uid} = \left\{D = g^{\frac{\beta+r}{\alpha}}\right\}$, and sends it to the user.

2) GENERATE USER PROXY KEY

The KA cooperates with attribute authorities to participate in the proxy key generation algorithm. This algorithm takes the user's attribute set as input and outputs the user proxy key.

i) The KA randomly selects $\varepsilon \in Z_p^*$, and computes $n = g^r$, $D' = g^r h^{\varepsilon}$, $D'' = g^{\varepsilon}$. Then the KA sends n to the relevant attribute authorities.

ii) The attribute authorities select a random number $r_j \in Z_p^*$ for the managed attribute λ_j, and compute $D_j = n \cdot H\left(\lambda_j\right)^{r_j}$, $D_j' = g^{r_j}$. Then they send $\left(D_j, D_j'\right)$ to the KA.

iii) Then the whole user proxy key is:

$$SK_{px,uid} = \left\{D' = g^r h^{\varepsilon}, D'' = g^{\varepsilon}, \forall \lambda_j \in S_1 : D_j = g^r \cdot H\left(\lambda_j\right)^{r_j}, D_j' = g^{r_j}\right\}.$$

The user proxy key will be updated in the next step.

3) GENERATE ATTRIBUTE GROUP KEY

The KA generates attribute groups $\left\{G_j, \forall \lambda_j \in S\right\}$. For example, if the attributes of $u1$, $u2$, $u3$ are $\{\lambda_1, \lambda_2, \lambda_3\}$, $\{\lambda_2, \lambda_3\}$, $\{\lambda_1, \lambda_3\}$ respectively, the KA generates $G_1 = \{u_1, u_3\}$, $G_2 = \{u_1, u_2\}$, $G_3 = \{u_1, u_2, u_3\}$. Then, the KA generates a group key for each attribute group as follows:

First, the leaf private key $c_{uid} \in Z_p^*$ is generated for each user in the group and sent to the user, the KA updates

$$SK_{px,uid} = \left\{D' = g^r h^{\varepsilon}, D'' = g^{\varepsilon}, \forall \lambda_j \in S_1 : D_j = g^r \cdot H\left(\lambda_j\right)^{r_j}, D_j' = \left(g^{r_j}\right)^{c_{uid}}\right\}.$$

and stores the updated user proxy key.

Then, a DH tree is constructed for each attribute as follows:

i) Each user's leaf private key is assigned to the leaf node of the tree.

ii) Calculate the non-leaf node private key. If the private keys of two children are x and y, the private key of this non-leaf node is $nk = \iota(g^{xy})$.

iii) Calculate the attribute group key. The attribute group key is the private key of the tree root.

Finally, the KA sends (uid, Hdr) to the CSP, where $Hdr = \left\{\forall \lambda_j \in S_1 : copath_j\right\}$. S_1 is a collection of a user's attributes, *copath* is the list of public keys of sibling nodes along its path. For example, the user u_0's *copath* is a collection of all the public keys of box nodes in Fig. 3. The red line is the path from the leaf node to the root node. To compute the attribute group key, we must know one leaf private key c_j, and all public keys in its *copath*. In addition, the number of leaf nodes in the tree must be 2^m, where $m \in N^*$. If the number of users in an attribute group is not 2^m, then the remaining leaf nodes are also assigned the elements in Z_p^*. When a new user joins the attribute group (the user obtains the attribute), one of the remaining leaf private keys is distributed to the new user.

C. ENCRYPTION

Before being uploaded to the CSP, a file M is processed with the following steps:

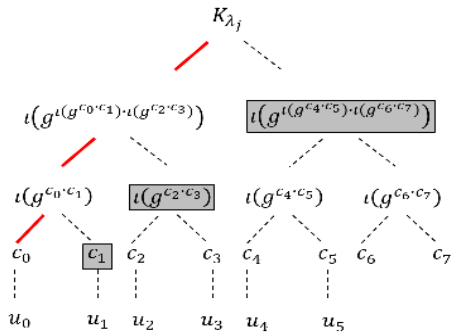


FIGURE 3. The *copath* of u_0 is a collection of public keys of box nodes.

i) The DO selects a content key ck at random, and encrypts M with ck by using the symmetric encryption algorithm. The result is denoted as $E_{ck}(M)$.

ii) The DO defines an access structure T, selects $s \in Z_p^*$ at random and computes $C = g^{\alpha s}$, $C_0 = g^s$, $C_1 = h^s$, then outputs the ciphertext as:

$$CT_1 = \left\{T, E_{ck}(M), \widetilde{C} = ck \cdot e(g,g)^{\beta s}, C = g^{\alpha s}, C_0 = g^s, C_1 = h^s\right\},$$

and sends CT_1 to the appointed fog node.

iii) After receiving CT_1, fog nodes perform attribute-related calculations for users. For the access tree T, fog nodes select a polynomial q_x for each node x in a top-down manner. The degree d_x of q_x is one less than the threshold value k_x, so we can get the value of d_x by $d_x = k_x - 1$.

Starting from the root node R, fog nodes select a random $s_1 \in Z_p^*$ and set $q_R(0) = s_1$, and then randomly choose d_R other points of q_R to define it completely. For any other node x, they set $q_x(0) = q_{parent(x)}(index(x))$, and then select d_x other points randomly to define q_x completely. Then the ciphertext CT is constructed as:

$$CT = \left\{T, E_{ck}(M), \widetilde{C}, C, C' = g^{s_1} \cdot g^s, C'' = h^{s_1} \cdot h^s, \left\{\forall y \in Y : C_y = g^{q_y(0)}, C'_y = H(\lambda_y)^{q_y(0)}\right\}\right\},$$

where Y is the set of attributes associating with leaf nodes of T.

Finally, fog nodes perform re-encryption. For all $G_y \in G$, fog nodes ask the KA for the attribute group key K_{λ_y} and then use it to re-encrypt CT to generate:

$$CT' = \left\{T, E_{ck}(M), \widetilde{C}, C, C' = g^{s_1} \cdot g^s, C'' = h^{s_1} \cdot h^s, \left\{\forall y \in Y : C_y = g^{q_y(0)}, C'_y = (H(\lambda_y)^{q_y(0)})^{K_{\lambda_y}}\right\}\right\}.$$

D. DECRYPTION

The decryption algorithm includes two sub-algorithms: a group key algorithm and a message decryption algorithm. The message decryption algorithm is further divided into two sub-algorithms: an outsourcing sub-algorithm performed by fog nodes and a local sub-algorithm performed by users. When a user needs to access data, the CSP sends CT' to the appointed fog node and sends Hdr to the user. Then the KA checks whether the user's proxy key is deleted. If not, the KA will send $SK_{px,uid}$ to the user.

1) GROUP KEY ALGORITHM

After receiving the head message Hdr from the CSP, the user uses his own leaf private key c_{uid} and the *copath* in the head message to calculate group keys of all attributes of him. If $u_t \in G_j$, then the user has a valid leaf private key c_t, and the group key is calculated by iteratively performing an exponentiation operation with the public keys in the *copath*. Then the user updates the proxy key with the leaf private keys and the attribute group keys as: $SK'_{px,uid} = \left\{D', D'', \forall \lambda_j \in S_1 : D_j, D'_j = (g^{r_j})^{\frac{c_{uid}}{K_{\lambda_j} \cdot c_{uid}}}\right\}$.

The user sends the updated proxy key $SK'_{px,uid}$ to the appointed fog node.

2) MESSAGE DECRYPTION ALGORITHM

This algorithm is divided into two parts: the decryption performed by fog nodes and the decryption performed by users.

i) Decryption performed by fog nodes.

Fog nodes define the recursive algorithm $Fog.DecryptNode(CT', SK'_{px,uid}, x)$ as:

a) If x is a leaf node. Let $\lambda_j = att(x)$, if $\lambda_j \notin S_1$, then $Fog.DecryptNode(CT', SK'_{px,uid}, x) = null$. Else, then:

$$\begin{aligned} &Fog.DecryptNode(CT', SK'_{px,uid}, x) \\ &= \frac{e(D_x, C_x)}{e(D'_x, C'_x)} \\ &= \frac{e(g^r \cdot H(\lambda_x)^{r_x}, g^{q_x(0)})}{e((g^{r_x})^{-K_{\lambda_x}}, (H(\lambda_x)^{q_x(0)})^{K_{\lambda_x}})} \\ &= e(g,g)^{rq_x(0)}. \end{aligned}$$

b) If x is a non-leaf node. For all children nodes z of x, fog nodes perform $F_z = Fog.DecryptNode(CT', SK'_{px,uid}, z)$. If less than k_x children nodes satisfy $F_z \neq null$, then $F_x = null$. Else, F_x is calculated as:

$$\begin{aligned} F_x &= \prod_{z \in S_x} F_z^{\Delta_{j,S'_x(0)}} \\ &= \prod_{z \in S_x} (e(g,g)^{rq_z(0)})^{\Delta_{j,S'_x(0)}} \\ &= \prod_{z \in S_x} (e(g,g)^{rq_{parent(z)}(index(z))})^{\Delta_{j,S'_x(0)}} \\ &= \prod_{z \in S_x} (e(g,g)^{rq_x(j)})^{\Delta_{j,S'_x(0)}} \\ &= e(g,g)^{rq_x(0)}, \end{aligned}$$

where $j = index(z)$ and $S'_x = \{index(z) : z \in S_x\}$.

Then fog nodes call functions on the root node R of T. If users' attribute collections S_1 satisfy T, fog nodes get:

$$\begin{aligned} F_R &= Fog.DecryptNode(CT', SK'_{px,uid}, R) \\ &= e(g,g)^{rq_R(0)} \\ &= e(g,g)^{rs_1}. \end{aligned}$$

Fog nodes also compute:

$$A' = \frac{e(D', C')}{e(D'', C'')} = \frac{e(g^r h^{\varepsilon}, g^{s_1} \cdot g^s)}{e(g^{\varepsilon}, h^{s_1} \cdot h^s)} = e(g, g)^{r(s_1+s)}$$

and $B' = A'/F_R = e(g, g)^{rs}$. Finally, fog nodes generates:

$$CT^p = \left\{E_{ck}(M), \widetilde{C} = ck \cdot e(g, g)^{\beta s}, C = g^{\alpha s}, B'\right\}.$$

ii) Decryption performed by users.

After receiving CT^p from fog nodes, users calculate ck as:

$$\frac{\widetilde{C} \cdot B'}{e(D, C)} = \frac{ck \cdot e(g, g)^{\beta s} \cdot e(g, g)^{rs}}{e(g^{\frac{\beta+r}{\alpha}}, g^{\alpha s})} = ck.$$

Thus, $E_{ck}(M)$ can be decrypted with ck by applying the symmetric decryption algorithm.

E. REVOCATION MECHANISM

1) USER REVOCATION

When the user revocation occurs, we do not need to update group keys and re-encrypt the ciphertext. The only thing we need to do is delete the revoked user's $SK_{px,uid}$ stored in the KA. Without $SK_{px,uid}$, the appointed fog node can no longer perform partial decryption for the revoked user. Therefore, the revoked user cannot decrypt the ciphertext.

2) ATTRIBUTE REVOCATION

When the attribute revocation occurs, the KA updates the attribute group and changes the attribute group key for the attribute which is affected by the membership change. Suppose the attribute λ_j of the user u_4 is revoked, as shown in Fig. 4, the proxy key and ciphertext is updated as follows:

i) The KA selects a new leaf private key c_4' for user u_4, then calculates the updated $copath'$ and the updated attribute group key K'_{λ_j}. Then the KA sends $\left\{\forall uid \in G(j) : \lambda_j, copath'\right\}$ and $\frac{K'_{\lambda_j}}{K_{\lambda_j}}$ to the CSP, but do not send the updated leaf private key c_4' to the user u_4. The KA updates the proxy key as:

$$SK_{px,uid} = \left\{D', D'', D_j = g^r \cdot H\left(\lambda_j\right)^{r_j}, D'_j = \left(g^{r_j}\right)^{c'_{uid}}, \forall \lambda_i \in S_1 \setminus \left\{\lambda_j\right\} : D_i = g^r \cdot H\left(\lambda_i\right)^{r_i}, D'_i = \left(g^{r_i}\right)^{Cuid}\right\}.$$

ii) For all users in G_j, the CSP updates their Hdr as $\left\{copath'_j, \forall \lambda_i \in S_1 \setminus \left\{\lambda_j\right\} : copath_i\right\}$, then selects a random $s' \in Z_p^*$ and updates ciphertext as:

$$CT' = \left\{T, E_{ck}(M), \widetilde{C} = ck \cdot e(g, g)^{\beta(s+s')}, C = g^{\alpha(s+s')}, C' = g^{(s_1+s')} \cdot g^{(s+s')}, C'' = h^{(s_1+s')} \cdot h^{(s+s')}, C_j = g^{q_j(0)+s'}, C'_j = (H(\lambda_j)^{q_j(0)+s'})^{K'_{\lambda_j}}, \forall y \in Y \setminus \{j\} : C_y = g^{q_y(0)+s'}, C'_y = (H(\lambda_y)^{q_y(0)+s'})^{K_{\lambda_y}}\right\}.$$

After that, when a user accesses for the data, the CSP responds with the updated ciphertext CT' and the updated head message Hdr.

The user u_4 cannot get the value of the updated leaf private key c_4', he cannot calculate the updated attribute group key, thus the attribute λ_j of u_4 is revoked. Moreover, the user u_4 cannot initiate a collusion attack with other users in the attribute group, because no user in the attribute group knows the value of the updated leaf private key c_4', whereas the user proxy key is bound to the latest version of the user leaf private key. Therefore, this revocation mechanism is anti-collusion attack.

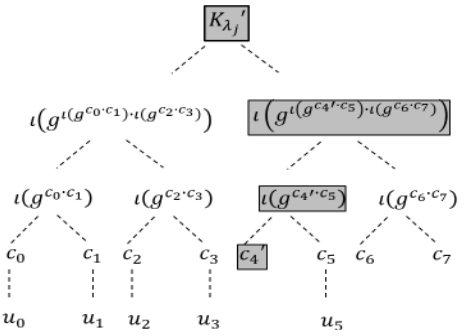


FIGURE 4. The attribute λ_j of u_4 is revoked.

VI. SECURITY ANALYSIS

In this section, we will analyze the security of the proposed scheme in terms of data confidentiality, collusion tolerance, forward security, and backward security.

• *The proposed scheme can achieve data confidentiality.* The proposed scheme can protect the confidentiality of outsourced data against illegal users. The attributes of illegal users can not satisfy the access policy in ciphertext, so they cannot recover the intermediate value $e(g, g)^{rs_1}$ in the decryption process. On the other hand, when a user is revoked, the KA will delete his proxy key. Without the proxy key, he could not recover the intermediate value either. At the same time, when the user revokes some attributes, he cannot decrypt the ciphertext unless his other attributes still satisfy the access policy. In order to decrypt the attribute λ_x, for node x, the user needs to pair the C'_x from the ciphertext with the D'_x from its proxy key. However, C'_x is bound to the attribute group key K_{λ_x}, so the user who has been revoked cannot calculate the value of $e(g, g)^{r \cdot q_x(0)}$, and he cannot recover the intermediate value $e(g, g)^{rs_1}$. In addition, since the KA, the CSP, the FN, and attribute authorities are semi-trusted, outsourced data should also be kept secret from them. In our scheme, the CSP issues the user a personalized private key by performing a secure 2PC protocol with the KA. This key generation protocol discourages the two parties to obtain each other's master private key. Therefore, the KA and the CSP cannot get enough information to decrypt the ciphertext. Although the KA and the FN can get the users' proxy keys, they can only obtain partially decrypted ciphertext. If there is no appointed user's private key, the KA and the FN can not further decrypt the ciphertext. And any attribute authority that attempts to forge the user proxy key will not succeed, because in the process of generating the attribute group key, the

KA will update the proxy key as:

$$SK_{px,uid} = \left\{D' = g^r h^{\varepsilon}, D'' = g^{\varepsilon}, \forall \lambda_j \in S_1 : D_j = g^r \cdot H\left(\lambda_j\right)^{r_j}, D'_j = \left(g^{r_j}\right)^{c_{uid}}\right\}.$$

During the decryption process, the user will update the proxy key as:

$$SK'_{px,uid} = \left\{D', D'', \forall \lambda_j \in S_1 : D_j, D'_j = (g^{r_j})^{\frac{c_{uid}}{K_{\lambda_j} \cdot c_{uid}}}\right\}.$$

Thus, in the decryption phase, the proxy key forged by the attribute authority cannot be successfully updated, and then the ciphertext cannot be successfully decrypted. Besides, even if the attribute authority can get the user's proxy key, it cannot decrypt the ciphertext successfully without appointed user's private key. Therefore, the data confidentiality against the semi-trusted CSP, FN, KA, and attribute authorities is also guaranteed.

• *The proposed scheme is resistant to the collusion attack.* The proposed scheme is also secure against the collusion attack. Suppose some collusive users can get all the attributes needed for decryption, they cannot successfully decrypt the ciphertext. Since when generating the private key, the KA selects different r for each user, and each user's $D_j = g^r \cdot H(\lambda_j)^{r_j}$ is different. As a result, each unauthorized user of all colluders can only calculate the value $e(g, g)^{rq_x(0)}$ of each corresponding node x, but can not jointly calculate $e(g, g)^{rs_1}$.

In addition, suppose $G_i = \{a, b\}$. When the attribute i of the user a is revoked, the attribute i of the user b is not revoked, user a and user b initiate a collusion attack. However, since neither user a nor user b knows the updated leaf private key c'_a, even if user a can request the updated attribute group key K'_{λ_i} of attribute i from user b, the user proxy key cannot be correctly updated as follows:

$$SK'_{px,uid} = \left\{D', D'', D_i = g^r \cdot H(\lambda_i)^{v_i}, D'_i = (g^{v_i})^{\frac{c'_a}{K'_{\lambda_i} \cdot c_a}}, \forall \lambda_j \in S \setminus \{i\} : D_j = g^r \cdot H(\lambda_j)^{v_j}, D'_j = (g^{v_j})^{\frac{c_a}{K_{\lambda_j} \cdot c_a}}\right\}.$$

As a result, the ciphertext cannot be decrypted correctly. Therefore, our scheme can resist against collusion attacks initiated by revoked users and valid users in the same attribute group.

• *The proposed scheme can achieve forward security and backward security.* The proposed scheme also guarantees backward security for users who obtain an attribute, and forward security for users who drop an attribute. In our scheme, when the user gets a new attribute, he and other valid attribute group members will get the updated attribute group key, and he will get the new attribute related key. The ciphertext components corresponding to the attributes C'_y will also be re-encrypted with the updated attribute group key, and all of the ciphertext components encrypted with the secret value s and s_1 will be re-encrypted by the CSP with a new random selected secret value s'. In this way, the previous ciphertext associated with the old attribute group key cannot be decrypted, even if the user stores the previous ciphertext. And if the user wants to decrypt $e(g, g)^{\beta s}$ from the current ciphertext and then decrypt the $\tilde{C}$ of the previous ciphertext to get ck, he will not succeed. Because the user can only decrypt $e(g, g)^{r(s+s')}$ from the current ciphertext, he can not further decrypt $e(g, g)^{\beta s}$, because the $e(g, g)^{\beta s}$ in the current ciphertext is protected by s'. Therefore, the forward security of outsourced data is guaranteed.

On the other hand, when a user drops an attribute (his remaining valid attributes no longer satisfy the access policy), the corresponding attribute group key will be updated and transmitted to other valid users excluding this user. The ciphertext components corresponding to the attributes will also be re-encrypted with the updated attribute group key, and all of the ciphertext components encrypted with the secret value s and s_1 will be re-encrypted by the CSP with a new random selected secret value s'. Because the user does not know the updated attribute group key, he can no longer decrypt the corresponding node of the attribute. Even if the user calculates the value of $e(g, g)^{\beta s}$ and stores it before revoking the attribute, he can not decrypt the updated ciphertext. Because $e(g, g)^{\beta s}$ of the updated ciphertext is protected by s'. Therefore, backward security of outsourced data can also be guaranteed.

TABLE 2. Symbols used in comparison.

Symbol	Explanation
$\|Z_p\|$	The bit length of the element in Z_p
$\|G\|$	The bit length of the element in G
$\|G_T\|$	The bit length of the element in G_T
$\|K\|$	The bit length of the KEK in scheme [10]
l	Number of attributes the user satisfies
n	Number of attributes associated with a user's private key
m	Number of attributes appeared in ciphertext
L	Number of attributes in the system
N	Number of users in each attribute group
M	Number of users in the system
T_1	Running time required for one exponentiation in G
T_2	Running time required for one exponentiation in G_T
T_e	Running time for one pairing operation

VII. PERFORMANCE ANALYSIS

In this section, we first summarize the functional features of some CP-ABE schemes in Table 3. Then, we compare our scheme with Zhang's scheme [8], Hur's scheme [10], Hur's improved scheme [17], and Chen's scheme [29] in terms of communication, storage overhead, and computation cost in Table 4, Table 5, and Table 6. Figure 5 shows our experimental results. The explanation of the symbols used in the above comparisons is shown in Table 2.

A. FUNCTIONAL FEATURES COMPARISON

Table 3 compares the functional features of some CP-ABE schemes in the cloud and fog computing.

As can be seen from Table 3, compared to Fan's scheme, Zhang's scheme outsources the complex encryption operations to fog nodes, further reduces the burden on users

TABLE 3. Functional features comparison.

Scheme	Fan's scheme [15]	Zhang's scheme [8]	Hur's scheme [10]	Hur's improved scheme [17]	Chen's scheme [29]	Our scheme
Environment	Fog	Fog	Cloud	Cloud	Fog	Fog
Encryption outsource	No	Yes	No	No	Yes	Yes
Decryption outsource	Yes	Yes	No	No	Yes	Yes
User revocation	Yes	No	No	No	Yes	Yes
Attribute revocation	No	Yes	Yes	Yes	Yes	Yes
Stateless attribute revocation	–	No	Yes	Yes	Yes	Yes
Against collusion attack	–	–	No	No	Yes	Yes
The KA is not fully trusted	No	No	No	Yes	No	Yes
Multi-authority	Yes	No	No	No	No	Yes

TABLE 4. Comparison of communication overhead.

Scheme		Zhang's scheme [8]	Hur's improved scheme [17]	Chen's scheme [29]	Our scheme
Encryption	DO-FN	$(m+2)\|G\|$	–	$3\|G\|+\|G_T\|$	$3\|G\|+\|G_T\|$
	DO-CSP	$(m+3)\|G\|+\|G_T\|$	$(2m+1)\|G\|+\|G_T\|$	–	–
	FN-CSP	–	–	$(2m+3)\|G\|+\|G_T\|$	$(2m+3)\|G\|+\|G_T\|$
Decryption	User-KA	–	–	–	$(2n+2)\|G\|$
	User-FN	$(n+3)\|G\|+2\|G_T\|$	–	$(2n+3)\|G\|+2\|G_T\|$	$(2n+3)\|G\|+2\|G_T\|$
	User-CSP	–	$(mN+2m+2)\|G\|+\|G_T\|$	$(mlog_2N+2n+2)\|G\|$	$mlog_2N\|G\|$
	FN-CSP	$(m+3)\|G\|+\|G_T\|$	–	$(2m+3)\|G\|+\|G_T\|$	$(2m+3)\|G\|+\|G_T\|$
Attribute revocation	KA-User	$M\cdot\|Z_p\|$	–	–	–
	KA-CSP	$\|Z_p\|$	–	–	$(log_2N+1)\|G\|$
	FN-CSP	–	–	$(log_2N+1)\|G\|$	–
User revocation	User-CSP	–	–	$\|Z_p\|$	–
	User-KA	–	–	–	$\|Z_p\|$

TABLE 5. Comparison of storage overhead.

Scheme	User	FN	CSP	KA
Zhang's scheme [8]	$(n+3)\|G\|$	–	$(m+3)\|G\|+\|G_T\|$	$L\|Z_p\|$
Hur's scheme [10]	$(2n+1)\|G\|+(log_2N+1)\|K\|$	–	$(2m+1)\|G\|+\|G_T\|+m(N/2)\|K\|$	$\|G\|+\|Z_p\|$
Hur's improved scheme [17]	$(2n+2)\|G\|$	–	$(mN+2m+3)\|G\|+\|G_T\|+\|Z_p\|$	$\|Z_p\|$
Chen's scheme [29]	$\|G\|+n\|Z_p\|$	$Mn\|Z_p\|$	$(Mn\cdot log_2N+2m+2n+5)\|G\|+\|G_T\|$	$\|G\|+\|Z_p\|$
Our scheme	$\|G\|+n\|Z_p\|$	0	$(Mn\cdot log_2N+2m+4)\|G\|+\|G_T\|$	$(2n+2)\|G\|+\|Z_p\|$

TABLE 6. Comparison of computation overhead on users.

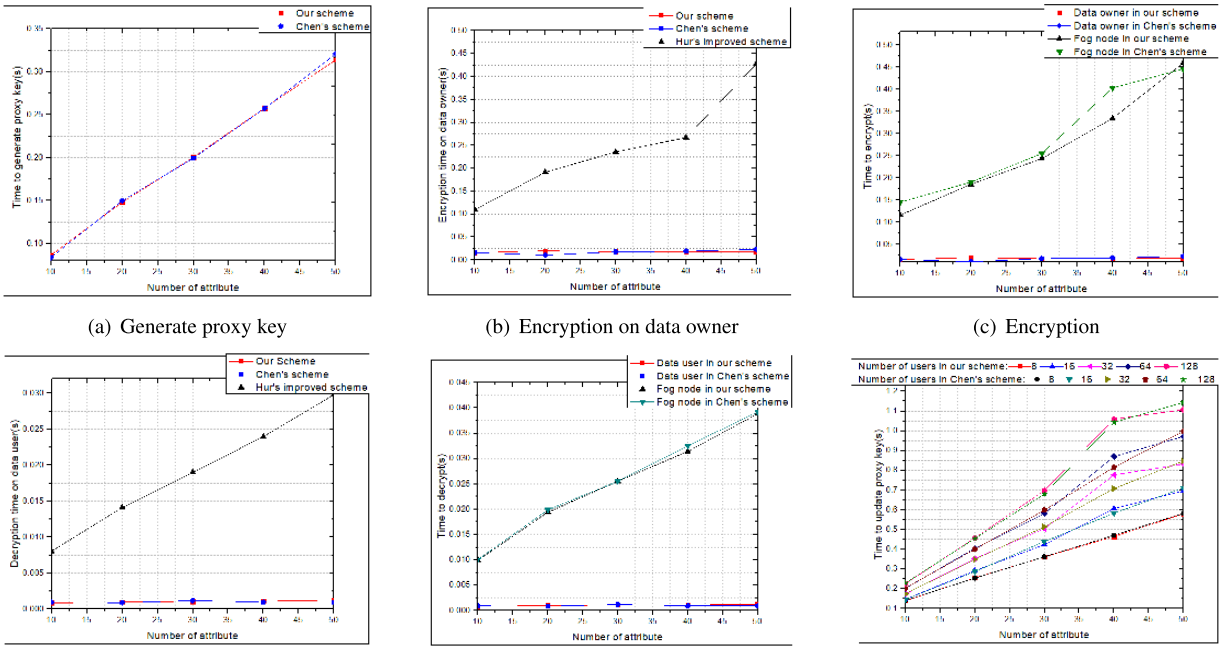
Scheme	Zhang's scheme [8]	Hur's scheme [10]	Hur's improved scheme [17]	Chen's scheme [29]	Our scheme
Encryption	$2T_1+T_2$	$(2m+1)T_1+T_2$	$(2m+1)T_1+T_2$	$3T_1+T_2$	$3T_1+T_2$
Decryption	T_e	$(2l+1)T_e$	$(2l+1)T_e$	T_e	T_e
Key update	T_1	nT_1	$n(N+1)T_1+nT_e$	$n\cdot(1+log_2N)T_1$	$n\cdot(1+log_2N)T_1$

and realizes a finer-grained attribute revocation. However Zhang's scheme requires users to update their private keys online from time to time, which does not meet the needs of the actual environment. Hur's scheme proposes a stateless attribute revocation scheme in the cloud, and Hur's improved scheme removes the trusted center in the stateless attribute revocable CP-ABE scheme. However, neither of these two schemes outsources the encryption and decryption operations, and they are not resistant to collusion attacks. Our newly proposed scheme achieves encryption and decryption outsourcing, user revocation, stateless attribute revocation, and effective resistance to collusion attack in the fog. At the same time, it has multiple authorities and removes the fully trusted authority center. Thus, our scheme is more efficient and available.

B. COMMUNICATION OVERHEAD

In Table 4, we theoretically compare the communication overhead of some CP-ABE schemes, which is mainly caused by the transmission of update messages, ciphertext, and head messages in the system.

It can be seen from Table 4 that the total communication overhead of Hur's improved scheme and our scheme is larger than that of Zhang's scheme, because Hur's improved scheme and our scheme introduce the head message *Hdr* to realize attribute revocation statelessly, so that users do not need to



(a) Generate proxy key (b) Encryption on data owner (c) Encryption

(d) Decryption on data owner (e) Decryption (f) Update proxy key

FIGURE 5. Comparison of computation overhead.

be online all the time to get the latest version of attribute group keys. And the total communication overhead of Hur's improved scheme is smaller than that of our scheme, because the network architecture of Hur's improved scheme is cloud, users communicate with the CSP directly. But in the phases of encryption and decryption, it will bring heavy computation costs to users.

In the encryption phase, the total communication overhead generated by all schemes is almost the same, and Chen's scheme and our scheme generate less communication overhead on the user side. But in the decryption phase, Hur's improved scheme, Chen's scheme, and our scheme cause more communication overhead due to the transmission of head messages. We note that Zhang's scheme and Hur's improved scheme cannot achieve user revocation, whereas Chen's scheme and our scheme cause very little communication overhead when realizing user revocation. As for attribute revocation, in Zhang's scheme, the KA must send update messages to the revoked user and the unrevoked user to update the attribute group keys. Frequent attribute revocation operations can cause both heavy communication overhead and computation cost. However, Hur's improved scheme, Chen's scheme and our scheme realize attribute revocation by updating the *Hdr*. There is no requirement for users to update the attribute group keys online. And when attribute revocation occurs, Hur's improved scheme does not generate communication overhead, because it relies on the cloud to complete the attribute revocation, which will cause an unbearable burden on the cloud. This is also the reason for introducing the fog node. Thus, Chen's scheme and our scheme are more suitable for the fog-based smart grid system where members change frequently.

Compared to Chen's scheme, our scheme does not increase the total communication overhead in the system at these phases, nor does it increase communication overhead on the user side. However, in order to remove the trusted center and increase the security of the scheme, our scheme needs the KA and the CSP to cooperate to generate the user private key, which increases the system communication overhead, but does not increase the communication overhead on the user side.

C. STORAGE OVERHEAD

In Table 5, we theoretically compare the storage overhead of some CP-ABE schemes, which is mainly caused by the storage of private keys, ciphertext, and head messages.

From Table 5, we can conclude that the storage overhead on users in Chen's scheme and our scheme is much less than that one in Zhang's scheme, Hur's scheme, and Hur's improved scheme. In these five schemes, the storage overhead on users is just the user private key and the private key used for attribute revocation. Whereas in Chen's scheme and our scheme, the user's private key is divided into two parts, the attribute-independent user's private key and the attribute-related proxy key. The user only stores the user's private key independent of the attribute, and the proxy key related to the attribute is stored by the CSP or the KA, so the storage overhead of the user in Chen's scheme and our scheme is reduced.

Whereas the storage cost of Zhang's scheme on the CSP is much less than that of Hur's scheme, Hur's improved scheme, Chen's scheme, and our scheme. This is because the CSP in Zhang's scheme only needs to store ciphertext, but the CSP in the other four schemes not only stores ciphertext but also stores head messages. However, Zhang's scheme and our scheme have a larger storage overhead than that of Hur's scheme, Hur's improved scheme, and Chen's scheme on the KA. Because in Zhang's scheme and our scheme, the KA needs to store an additional attribute version key or proxy key, whereas in the other three schemes, the KA only needs to store its own master private key.

Compared with Chen's scheme, our scheme generates the same storage overhead on users. Although our scheme produces less storage overhead on the CSP, it produces larger storage overhead on the KA. And our scheme and Chen's scheme generate the same total storage overhead on the KA and the CSP. In addition, Chen's scheme also incurs large storage overhead on fog nodes, which is improved by our scheme. Our scheme resists collusion attack by binding the user's leaf private key to the user's private key, eliminates the need for fog nodes to store the leaf private key for all attributes of all users. So, in general, our scheme generates less storage overhead in the system than Chen's scheme.

D. COMPUTATION COST

In the actual environment, the resources of users are subject to certain restrictions, whereas fog nodes and the CSP are more powerful. Therefore, in Table 6, we compare the users' computation cost of Zhang's scheme, Hur's scheme, Hur's improved scheme, Chen's scheme, and our scheme in encryption, decryption, and key update phases. In these schemes, we only consider pairing operations and exponentiation operations, since the other operations are too fast compared to pairing operations.

It can be seen from Table 6 that the computation cost of the encryption and decryption in Hur's scheme and Hur's improved scheme is much larger than that of Zhang's scheme, Chen's scheme, and our scheme. This is because Zhang's scheme, Chen's scheme and our scheme outsource the complex operations from users to fog nodes, which reduces the burden on data owners and users. In addition, Hur's scheme, Hur's improved scheme, Chen's scheme, and our scheme generate more computation cost than Zhang's scheme when updating the private key. To revoke attribute statelessly, Hur's scheme, Hur's improved scheme, Chen's scheme, and our scheme implicitly store the updated attribute group key in the head message. And users need to recover the attribute group key from the head message after receiving it, which causes more computation cost. Whereas in Zhang's scheme, when attribute revocation occurs, each user only needs to perform an exponential operation, but all users must update the private key online from time to time. Although the computation overhead incurred by each user is not large, Zhang's scheme imposes a large computation overhead on the system, especially when there are massive users in the system. Besides, the computation overhead generated by Chen's scheme and our scheme at these phases is exactly the same. However, in the key generation phase, the KA and the CSP in our scheme will participate in a 2PC protocol, which will incur more computation overhead.

Furthermore, we compare the efficiency of Hur's improved scheme, Chen's scheme, and ours in experimental aspect. We perform these schemes on a Windows 10 system with an Inter(R) Core(TM) i5-7200 CPU at 2.50 GHz and 8.00 GB RAM. Both schemes are run by applying the cpabe toolkit [30] and the Pairing-Based Cryptography library. The number of attributes used in the experiments is $L = \{10, 20, 30, 40, 50\}$.

Figure 5 shows the computation time of proxy key generation, encryption, decryption, and key update in Hur's improved scheme, Chen's scheme, and our scheme. Figure 5(a) gives the comparison of the computation time of proxy key generation in Chen' scheme and our scheme. We find that the computation time of proxy key generation in both schemes is almost same. Figure 5(b),(d) show the comparison of encryption and decryption time in these three schemes respectively. We can observe that the encryption (decryption) time of data owners (users) in Hur's improved scheme is longer than that in Chen's scheme and our scheme, and the difference becomes more and more obvious with the increase of the number of attributes. Because data owners (users) in Hur's improved scheme need to perform attribute-related computation, whereas Chen's scheme and our scheme outsource all operations related to attributes to fog nodes. Figure 5(c),(e) show the comparison of encryption and decryption time on data owners and fog nodes in Chen's scheme and our scheme, respectively. These also show that Chen's scheme and our scheme outsource complex calculations to fog nodes, leaving only a small amount of constant calculations for users. The reason we outsource the calculations of users to fog nodes is that user resources are limited and fog nodes have more powerful processing capabilities.

Figure 5(f) simulates the time of calculating attribute group keys and updating proxy keys when the number of users in each attribute group is $U = \{8, 16, 32, 64, 128\}$ in Chen's scheme and our scheme. When users receive the head message sent by the CSP, they use the leaf private key and *copath* in the head message to calculate the attribute group key. Then users update their proxy keys with the updated group keys. This operation only occurs when users request data stored in the CSP. In fact, users do not need to perform any key update operation when users or attributes are revoked continuously. In a word, Chen's scheme and our scheme add a little overhead to achieve attribute revocation. And as shown in Fig. 5, Chen's scheme and our scheme have the same computation overhead for users in these three phases. In order to remove the fully trusted key authority center, our scheme will add a little overhead in user private key generation phase.

VIII. CONCLUSION

In this paper, we have proposed a revocable multi-authority CP-ABE scheme without the trusted authority center in the fog-based smart grid system, which outsources the attribute-related calculations to fog nodes. The proposed scheme uses a secure 2PC protocol between the KA and the CSP to generate the user private key, and none of the two parties can compute the user's private key alone. The DH tree is used to revoke attribute statelessly for the first time, and solves the problem that Hur's scheme and Hur's improved scheme cannot resist collusion attack. In addition, our scheme outsources the complex operations to fog nodes, and combines two granular revocation mechanisms, so that more resource-limited devices can join the fog-based smart grid system. What's more, user's proxy key is not only related to attribute group key, but also to user's leaf private key, which not only improves the security of the scheme, but also reduces the storage overhead of the system. Security analysis shows that our scheme can achieve confidentiality, forward and backward security, and can resist collusion attack. From the performance evaluation, it can be seen that the users' encryption and decryption cost in our scheme are relatively low.

REFERENCES

[1] Z. Guan, J. Li, L. Wu, Y. Zhang, J. Wu, and X. Du, "Achieving efficient and secure data acquisition for cloud-supported Internet of Things in smart grid," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1934–1944, Dec. 2017.

[2] T. D. Dang and D. Hoang, "A data protection model for fog computing," in *Proc. 2nd Int. Conf. Fog Mobile Edge Comput. (FMEC)*, May 2017, pp. 32–38.

[3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput.*, Aug. 2012, pp. 13–16.

[4] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: Privacy preserving fog-enabled aggregation in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3733–3744, Aug. 2018.

[5] S. Zeadally, A.-S. K. Pathan, C. Alcaraz, and M. Badra, "Towards privacy protection in smart grid," *Wireless Pers. Commun.*, vol. 73, no. 1, pp. 23–50, Nov. 2013.

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

[7] Y. Xue, K. Xue, N. Gai, J. Hong, D. S. L. Wei, and P. Hong, "An attribute-based controlled collaborative access control scheme for public cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 11, pp. 2927–2942, Nov. 2019.

[8] P. Zhang, Z. Chen, J. K. Liu, K. Liang, and H. Liu, "An efficient access control scheme with outsourcing capability and attribute update for fog computing," *Future Generat. Comput. Syst.*, vol. 78, pp. 753–762, Jan. 2018.

[9] W. Teng, G. Yang, Y. Xiang, T. Zhang, and D. Wang, "Attribute-based access control with constant-size ciphertext in cloud computing," *IEEE Trans. Cloud Comput.*, vol. 5, no. 4, pp. 617–627, Oct./Dec. 2017.

[10] 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, Jul. 2011.

[11] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2005, pp. 457–473.

[12] 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.*, Nov. 2006, pp. 89–98.

[13] M. Chase, "Multi-authority attribute based encryption," in *Proc. Conf. Theory Cryptogr.* Berlin, Germany: Springer, 2007, pp. 515–534.

[14] D. Ramesh and R. Priya, "Multi-authority scheme based CP-ABE with attribute revocation for cloud data storage," in *Proc. Int. Conf. Microelectron., Comput. Commun. (MicroCom)*, Jan. 2016, pp. 1–4.

[15] K. Fan, J. Wang, X. Wang, H. Li, and Y. Yang, "Secure, efficient and revocable data sharing scheme for vehicular fogs," *Peer-to-Peer Netw. Appl.*, vol. 11, no. 4, pp. 766–777, Jul. 2018.

[16] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. ACM Conf. Comput. Commun. Secur.*, Nov. 2009, pp. 121–130.

[17] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2271–2282, Oct. 2013.

[18] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proc. ACM Conf. Comput. Commun. Secur.*, Oct. 2008, pp. 417–426.

[19] M. Pirretti, P. Traynor, and P. McDaniel, "Secure attribute-based systems," *J. Comput. Secur.*, vol. 18, no. 5, pp. 799–837, 2010.

[20] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proc. ACM Symp. Inf., Comput. Commun. Secur.*, Apr. 2010, pp. 261–270.

[21] K. Zhang, J. Gong, S. Tang, J. Chen, X. Li, H. Qian, and Z. Cao, "Practical and efficient attribute-based encryption with constant-size ciphertexts in outsourced verifiable computation," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, May 2016, pp. 269–279.

[22] Z. Zhou and D. Huang, "Efficient and secure data storage operations for mobile cloud computing," in *Proc. Int. Conf. Netw. Service Manage.*, Oct. 2012, pp. 37–45.

[23] M. Asim, M. Petkovic, and T. Ignatenko, "Attribute-based encryption with encryption and decryption outsourcing," in *Proc. 12th Austral. Inf. Secur. Manage. Conf.* Perth, WA, Australia: ECU Security Research Institute, Dec. 2014, pp. 21–28.

[24] S.-Y. Tan, "Comment on 'Secure data access control with ciphertext update and computation outsourcing in fog computing for Internet of Things,'" *IEEE Access*, vol. 6, pp. 22464–22465, 2018.

[25] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.

[26] K. Cohn-Gordon, C. Cremers, L. Garratt, J. Millican, and K. Milner, "On ends-to-ends encryption: Asynchronous group messaging with strong security guarantees," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 1802–1819.

[27] S. S. M. Chow, "Removing escrow from identity-based encryption," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2009, pp. 256–276.

[28] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham, "Randomizable proofs and delegatable anonymous credentials," in *Proc. Int. Cryptol. Conf. Adv. Cryptol.* Berlin, Germany: Springer, 2009, pp. 108–125.

[29] S. Chen, M. Wen, R. Lu, J. Li, and S. Chen, "Achieve revocable access control for fog-based smart grid system," in *Proc. IEEE 90th Veh. Technol. Conf.*, Sep. 2019, pp. 1–7.

[30] J. Bethencourt, A. Sahai, and B. Waters. (2011). *Advanced Crypto Software Collection: The Cpabe Toolkit*. [Online]. Available: http://acsc.cs.utexas.edu/cpabe/index.html

MI WEN received the M.S. degree in computer science from the University of Electronic Science and Technology of China, in 2005, and the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2008. From May 2012 to May 2013, she was a Visiting Scholar with the University of Waterloo, Waterloo, ON, Canada. She is currently a Professor with the College of Computer Science and Technology, Shanghai University of Electric Power. Her research interests include privacy preserving in wireless sensor network and smart grid. She has been a TPC Member of some flagship conferences, such as the IEEE INFOCOM, the IEEE ICC, and the IEEE GLOEBECOM, since 2012. She serves as an Associate Editor of *Peer-to-Peer Networking and Applications* (Springer).

SHAN CHEN received the B.E. degree (Hons.) from the College of Computer Science and Technology, Shanghai University of Electric Power, China, in 2017, where she is currently pursuing the M.S. degree. She has published Achieve Revocable Access Control for Fog-Based Smart Grid System. Her research interests include information security and smart grid.

RONGXING LU received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2012, where he was a Postdoctoral Fellow, from May 2012 to April 2013. He was an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore, from April 2013 to August 2016. He has been an Assistant Professor with the Faculty of Computer Science (FCS), University of New Brunswick (UNB), Fredericton, NB, Canada, since August 2016. He has published extensively in his areas of expertise. His research interests include applied cryptography, privacy-enhancing technologies, and the IoT-big data security and privacy. He is currently a Senior Member of the IEEE Communications Society. He was a recipient of the most prestigious Governor General's Gold Medal, during his Ph.D. degree, and eight best (student) article awards from some reputable journals and conferences. He was also a recipient of the 8th IEEE Communications Society (ComSoc) Asia Pacific (AP) Outstanding Young Researcher Award, in 2013. He is the Winner of the 2016–17 Excellence in Teaching Award, FCS, UNB. He currently serves as the Vice-Chair (Publication) of the IEEE ComSoc Communications and Information Security Technical Committee (CIS-TC).

BEIBEI LI received the B.E. degree (Hons.) in communication engineering from the Beijing University of Posts and Telecommunications, China, in 2014, and the Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2019. He was invited as a Visiting Researcher with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada, from March to August, 2018, as well as the Research Group of NEtworked Sensing and Control (NESC), College of Control Science and Engineering, Zhejiang University, China, from February to April 2019. He is currently an Associate Professor with the College of Cybersecurity, Sichuan University, China. His research studies have been published in the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, ACM *Transactions on Cyber-Physical Systems*, the IEEE INTERNET OF THINGS JOURNAL, *Information Sciences*, the IEEE GLOBECOM, and the IEEE ICC. His research interests include cyber-physical system security, with a focus on intrusion detection techniques, applied cryptography, and big data privacy in smart grids and industrial control systems. He served as a TPC Member for several international conferences, including the IEEE GLOBECOM, WCSP, and ICNC. He was a recipient of the Full Research Scholarship, during his Ph.D. degree.

SIJIA CHEN received the B.E. degree (Hons.) from the College of Information and Electrical Engineering, Shenyang Agricultural University, China, in 2017. She is currently pursuing the M.S. degree with the College of Computer Science and Technology, Shanghai University of Electric Power, China. She has published Data Deduplication in Fog Storage. Her research interests include deduplication and fog computing.