IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 30, 2018, accepted December 27, 2018, date of publication January 4, 2019, date of current version February 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2890976

Efficient CCA2 Secure Flexible and Publicly-Verifiable Fine-Grained Access Control in Fog Computing

DAWEI LI[1], JIANWEI LIU[2], QIANHONG WU[2], (Member, IEEE), AND ZHENYU GUAN[2], (Member, IEEE)

[1]School of Electronic and Information Engineering, Beihang University, Beijing 100191, China
[2]School of Cyber Science and Technology, Beihang University, Beijing 100191, China

Corresponding author: Zhenyu Guan (guanzhenyu@buaa.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0802502, in part by the National Natural Science Foundation of China under Grant 61672083, Grant 61370190, Grant 61772538, Grant 61532021, Grant 61472429, Grant 61402029, and Grant 61702028, in part by the Foundation of Science and Technology on Information Assurance Laboratory under Grant 61421120305162112006, and in part by the National Cryptography Development Fund under Grant MMJJ20170106.

ABSTRACT Fog computing enables computation, storage, applications, and network services between the Internet of Things and the cloud servers by extending the Cloud Computing paradigm to the edge of the network. When protecting information security in Fog computing, advanced security with low latency, wide-spread geographical distribution support, and high flexibility should be taken in to considertion first, because of its huge number of nodes. In this paper, we propose a new cryptographic primitive, named CCA2 secure publicly-verifiable revocable large-universe multi-authority attribute-based encryption (CCA2-PV-R-LU-MA-ABE), to achieve flexible fine-grained access control in Fog computing. In this primitive, end nodes in fogs generate private keys from multiple authorities that might be differentiated by their geographical locations or functions, and their attributes can be denoted by any strings in the large universe, which meets diverse needs in practical Fog applications. In addition, the accessibility of nodes can be revoked efficiently even by resource-limited devices. To ensure the validity of ciphertext, this primitive supports public verification and only valid ciphertext can be stored or transmitted. Based on the primitive and the feature of Fog computing, we construct a concrete CCA2-PV-R-LU-MA-ABE scheme. We define the security model of this primitive, which is much more secure than the CPA-secure scheme. Finally, we compare the efficiency of the proposed concrete scheme with that of the existing CPA-secure scheme by both theoretical and experimental analysis, and the results show that the extra consumption of efficiency to improving CPA to CCA2 is considerably low. The proposed scheme is highly secure, flexible, and efficient enough to be deployed in practical Fog computing.

INDEX TERMS Fog computing, attribute-based encryption, distributed access control, multi-authority, large-universe.

I. INTRODUCTION

Cloud Computing seems to provide an ideal solution for the processing of large amounts of data in Internet of Things (IoT), in which end users upload request and download result from a cloud center. And with the rapid development of Information-Centric IoT (IC-IoT), Device-to-Device (D2D) communication makes it more convenient to be deployed [1]. However, this paradigm is not suitable in some circumstances, like latency-sensitive applications in wireless access sensor network or in mobile equipment. To fill this gap, Fog Computing extends Cloud Computing to the edge of the network to provide low latency and location awareness for streaming and realtime applications in IoT. Source-limited end-user devices like wireless sensors are implemented in Fogs to achieve realtime smart computation in connected vehicles, smart grid, wireless sensors and actuators networks, etc. Fog Computing has a wide range of application scenarios, but the problem of information security follows. However, to protect information security in Fogs, existing cryptography schemes in Cloud Computing cannot be

applied to Fogs directly, considering efficiency and security level.

Attribute-Based Encryption (ABE) [2] is a suitable cryptosystem to achieve fine-grained access control. Users generate private keys corresponding to their attributes from an authority. Data owners can encrypt data according to the target recipients' attributes without knowing their exact identities, then only someone satisfying requested attributes can get access to the message. However, end-user devices in Fogs are distributed geographically, which is different from centralized Cloud, and sensors in Fogs have all kinds of attributes. Apparently, billions of end nodes with different attributes from their geographically isolated Fogs get private keys from a centralized attribute authority cannot meet the Fogs' requirement, especially in efficiency [3]. Besides, attributes might be various and expressed as all kinds of strings. Thus the attribute universe should be large enough to make the system flexible.

Then many large-universe multi-authority ABE (LU-MA-ABE) schemes was proposed to make ABE more flexible, while these scheme are constructed based on composite-order bilinear groups, which have unsatisfactory computational efficiency. Recently, Rouselakis and Waters [4] proposed an efficient LU-MA-ABE on prime-order bilinear groups, and it has great flexibility and efficiency. However, the scheme is only statically secure against chosen plaintext attack (CPA) that weaker than semantic security against chosen ciphertext attacks (CCA2). In traditional way, one-time signature is applied to improve CPA security to CCA2 at the cost of much more operation time. How to improve the efficient CPA-secure LU-MA-ABE to CCA2-secure LU-MA-ABE at the expense of smaller efficiency losses is a difficult problem to be solved urgently. For the limited source of end nodes, tampered or illegally encrypted message would cause a meaningless latency to Fogs, in which case a public verification mechanism is really needed to filter it.

In addition, the huge number of nodes in Fogs poses a problem for management. Once one of the billions of nodes leaks its private key, behaves illegally or its attributes change, the authority should revoke its private key. In the existing LU-MA-ABE scheme, the authorities has to update keys for all unrevoked nodes one by one, or perform complex calculations to unrevoked nodes, which will sharply reduce the efficiency of Fogs or increase energy consumption of devices in practice. In summary, an efficient LU-MA-ABE with CCA2 security supporting efficient private-key revocation should be presented to achieve efficient, secure and practical fine-grained access control in Fog Computing.

A. OUR CONTRIBUTIONS

We propose a new efficient CCA2-PV-LU-MA-ABE system for Fog Computing based on an earlier version [5] of this paper presented at the international conference, Cyberspace Safety and Security - 9th International Symposium [6], in which CCA2 secure is more secure than the exiting CPA secure ABE scheme and we add to it the function of public verification. The proposed system supports multiple authority and large-universe attributes to make the access control more flexible. Besides, this system supports public verification of ciphertext to ensure its validity, where without knowing the decryption keys, the Fogs and all end nodes can verify the validation of the ciphertext, i.e., the ciphertext is encrypted by the declared attributes and has not been tampered. This verification procedure prevents useless data from occupying storage space, which can be efficiently computed by fogs. This system satisfies the functionality, security and efficiency needs of fine-grained access control in Fog Computing.

We define the security notion of this system named indistinguishability against selective authority and access policy and statically chosen ciphertext attacks (IND-sAA-sCCA2). In this security notion, the challenger plays a security game with an adversary. At the beginning of the game, the adversary should declare the access policy and a set of corrupted authority that it will attack. Then the adversary plays a statically CCA2 security game with the challenger, but it cannot distinguish the encrypted message with a more than negligible probability.

Based on system model and security notion, we construct a concrete CCA2 secure PV-R-LU-MA-ABE scheme on prime-order groups. We use the subset-cover revocation framework to make the attribute revocable, and it reduces the number of executions in updating keys from linearly to logarithmically correlate with the number of end nodes. After revocation, the Fogs only need to re-encrypt the corresponding part of ciphertext to prevent end nodes with revoked attribute cannot get access to the data. To satisfy the CCA2 security need of the security notion, we use the Chameleon hash function, and we treat the hash value of the ciphertext as an on-the-fly dummy attribute. Then we use the hash-value attribute to encrypt the message and get a new part of ciphertext to make sure the ciphertext tamper-resistant. And the challenger can use the trapdoor of the Chameleon hash to complete the simulation of CCA2 security game.

Finally, we prove the proposed CCA2-PV-R-LU-MA-ABE scheme is IND-sAA-sCCA2 secure and we analysis the efficiency of this scheme. Above all, the CCA2-PV-R-LU-MA-ABE can satisfy almost all of the functionality, security and efficiency needs of Fog Computing, and thus it would get practical deployment in Fogs.

B. RELATED WORK

Fog Computing was proposed to enable data processing at the edge of the Cloud [7]. Since then, a lot of research on fog computing has been studied [8]–[11]. Hong *et al.* [12] proposed the mobile fog to achieve large-scale applications on the Internet of Things. Then Stojmenovic and Wen [13] and Stojmenovic *et al.* [14] introduced the scenarios and security issues of Fog Computing, and they studied the CPU and memory consumption on Fog devices when a man-in-the-middle attack happens. Jiang *et al.* [15] proposed a traceable ciphertext-policy ABE scheme in Fog Computing and it

provides protections against the key-delegation abuse issue. Zhang *et al.* [16] proposed CP-ABE scheme with outsourcing capability and attribute update for fog computing.

Sahai and Waters [2] first introduced ABE, and then key-policy ABE (KP-ABE) was proposed by Goyal *et al.* [17] and ciphertext-policy ABE (CP-ABE) was proposed by Bethencourt *et al.* [18]. In KP-ABE, secret keys are associated with access policy and ciphertexts are associated with attributes. In CP-ABE the ciphertexts are associated with access policy and secret keys are associated with attributes. To enhance the security, many ABE scheme was proposed [19]–[22]. Okamoto and Takashima [23] and Lewko *et al.* [24] proposed fully secure ABE constructions in standard model. MA-ABE was proposed by Chase [3] and it has been improved in both efficiency and security [25]–[29].

LU-ABE was proposed by Lewko *et al.* based on composite-order bilinear groups [30] and prime-order bilinear groups [31]. Rouselakis and Waters [32] proposed LU-ABE with selective security in standard model. Key revocation of ABE is constructed in direct and indirect methods [33]. Beimel *et al.* [34] proposed the revocable KP-ABE in the indirect method by updating keys. Sahai *et al.* [35] proposed the revocable ABE scheme by updating keys and ciphertext. And Tsuchida *et al.* [36] came up with the CPA secure revocable MA-ABE scheme.

To achieve CCA2 secure ABE scheme, Canetti *et al.* [37] proposed the approach that CPA-secure KP-ABE in [17] can be converted to CCA2-secure KP-ABE. Yamada *et al.* [38] proposed a generic construction that can transform ABE scheme from CPA security to CCA2 security, if the ABE scheme satisfies delegatability or verifiability. The above conversion approach is with the help of one-time signatures. Chen *et al.* [39] and Ge *et al.* [40] constructed ABE schemes without one-time signature, but the limitation of these scheme is that only threshold access policies are supported. Recently, Liu *et al.* [41] proposed CCA2-secure KP-ABE based on the Chameleon hash function.

C. ORGANIZATION

Necessary preliminaries are introduced for RHIBBE in Section II, including notations, the prime-order bilinear groups, the q-DPBDHE2 problem and assumption, access structures and linear secret sharing schemes, Chameleon hash function, and subset-cover revocation framework. Then the CCA2-PV-R-LU-MA-ABE system definition is presented in Section III, where the security notion IND-sAA-sCCA2 are described. Based on all of the above, a concrete CCA2-PV-R-LU-MA-ABE scheme is constructed in Section IV. Then the security of the scheme is analyzed in Section V and the proposed concrete scheme is IND-sAA-sCCA2 secure. In Section VI, the performance of the scheme is analyzed with both theoretical and experimental methods, which shows this system is suit for real deployment in Fog Computing to provide highly secure fine-grained access control. Finally, we give a future implementation example in Blockchain in Section VII and the proposed scheme would be efficiently deployed in blockchain-based distributed storage system.

TABLE 1. Notations.

Notation	Description
λ	Security Parameter
ϵ	Negligible Function of λ
$[n]$	$\{1, 2, \ldots n\}$
$\|S\|$	Cardinality of S
GID	Identity of the node
$\mathbb{Z}_p^{m\times n}$	Matrix with m rows and n columns elements in $\mathbb{Z}_p$
$\overrightarrow{v}\cdot\overrightarrow{\omega}$	Inner product of two vectors

II. PRELIMINARIES

A. NOTATIONS

CCA2-PV-R-LU-MA-ABE is a new cryptographic primitive for fog computing. To describe it much better, we introduce several notations in this paper. Table 1 summarizes these notations and their corresponding meanings.

We use $[n]$ to denote $\{1, 2, \ldots, \mathrm{n}\}$, and $|S|$ to denote the cardinality of the set S. And $s \xleftarrow{R} S$ denotes that s is uniformly chosen at random from S. Besides, we use $\mathbb{Z}_p^{m\times n}$ to denote a matrix with m rows and n columns elements in $\mathbb{Z}_p$. The inner product of two vectors is denoted as $\overrightarrow{v}\cdot\overrightarrow{\omega}$.

B. BILINEAR GROUPS AND COMPUTATIONAL ASSUMPTION

Let p be a large prime, and $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of order p. If g is a generator of $\mathbb{G}$, and $e : \mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ is a bilinear map, we say that $\mathbb{G}$ and $\mathbb{G}_T$ are bilinear groups if e satisfies all of the properties:

- Bilinearity: $\forall u, v \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab} = e(u^b, v^a)$.
- Non-degeneracy: $e(g, g) \neq 1$.
- Computability: group operation $e(u, v)$ for $u, v \xleftarrow{R} \mathbb{G}$ can be efficiently computed.

1) q-DPBDHE2 PROBLEM [4]

Let $\mathbb{G}$ and $\mathbb{G}_T$ be bilinear groups with order p, $e : \mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$, and let g be a generator of the group $\mathbb{G}$. Pick $s, a, b_1, b_2, \ldots, b_q \xleftarrow{R} \mathbb{Z}_p$ and $R \xleftarrow{R} \mathbb{G}_T$ and a tuple

$$D = (\mathbb{G}, p, e, g, g^s, \{g^{a^i}\}_{\substack{i\in[2q],\\ i\neq q+1}}, \{g^{b_j a^i}\}_{\substack{(i,j)\in[2q,q],\\ i\neq q+1}}, \{g^{s/b_i}\}_{i\in[q]}, \{g^{sa^i b_j/b_{j'}}\}_{\substack{(i,j,j')\in[q+1,q,q]\\ j\neq j'}})$$

Given the tuple $(D, e(g, g)^{sa^{q+1}})$ or (D, R), an algorithm $\mathcal{B}$ outputs a bit $b \in \{0, 1\}$. We define that $\mathcal{B}$ has advantage ϵ in solving q-DPBDHE2 problem if

$$|\Pr[\mathcal{B}(D, e(g, g)^{sa^{q+1}}) = 0] - \Pr[\mathcal{B}(D, R) = 0]| \geq \epsilon$$

where the probability is over the random choice of s, $a, b_1, b_2, \ldots, b_q \in \mathbb{Z}_p$, $g \in \mathbb{G}, R \in \mathbb{G}_T$ and random bits used in $\mathcal{B}$.

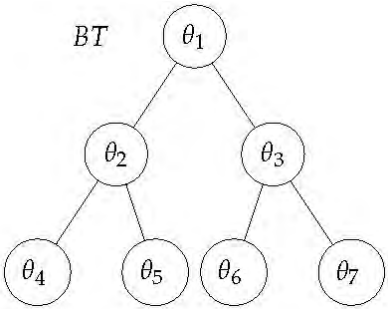

FIGURE 1. Binary tree BT.

2) q-DPBDHE2 Assumption.

The q-DPBDHE2 assumption holds in $\mathbb{G}$ if there is no algorithm that has advantage at least ϵ in solving the (t, ϵ, q)-DPBDHE2 problem in polynomial time t.

C. CHAMELEON HASH

In this paper, we use the Chameleon hash function to help prove the security of the concrete scheme. In the Chameleon hash function, there are three algorithms and only the one knowing the secret key can find collisions for its inputs.

- $KeyGen_{ch}(1^{\lambda}) \rightarrow (SK_{ch}, PK_{ch})$: This algorithm takes the security parameter $\lambda \in \mathbb{N}$ as input and then outputs the secret key and the public key.
- $H_{ch}(PK_{ch}, M, r_{ch}) \rightarrow V$: This algorithm takes the public key PK_{ch}, the message M and an auxiliary parameter r_{ch} as input and then outputs the hash value V.
- $UForge_{ch}(SK_{ch}, M, r_{ch}, M') \rightarrow r'_{ch}$: This algorithm takes the secret key SK_{ch}, a message M with its auxiliary parameter r_{ch} and a new message M' as input and then outputs a new auxiliary parameter r'_{ch} such that the hash value of M with r_{ch} is equal to the hash value of M' with r'_{ch}, i.e., $H_{ch}(PK_{ch}, M, r_{ch}) = H_{ch}(PK_{ch}, M', r'_{ch})$.

Besides, a Chameleon hash function should satisfy the security requirements including collision resistance and uniformity. The collision resistance means that, for a Chameleon hash public key PK_{ch}, there is no efficient algorithm that can find two pairs (M, r_{ch}), (M', r'_{ch}) where $M \neq M'$, such that $H_{ch}(PK_{ch}, M, r_{ch}) = H_{ch}(PK_{ch}, M', r'_{ch})$ except negligible probability. And the uniformity means that all messages M induce the same probability distribution on $H_{ch}(PK_{ch}, M, r_{ch}) \rightarrow V$ where r_{ch} is uniformly chosen at random.

D. SUBSET-COVER REVOCATION FRAMEWORK

Naor *et al.* [42] proposed the subset-cover revocation framework, in which the Complete Subtree (CS) method and Subset Difference (SD) methods are two instances. Based on CS method, we realize the revocation in this paper. The binary tree of CS method is described as the Figure 1 shows [43].

On input a binary tree BT, the current time T, and a revocation list RL, the algorithm outputs a set of users which has not been revoked until time T. What's more important, this set allows to update keys for the least nodes, which is logarithmic in the number of users. In the binary tree BT, each user is assigned as a leaf node of the binary tree, and the intermediate nodes of the binary tree are virtual points. Let v denote a non-leaf node and v_L (v_R) denote the left (right) child of v. Path(v_i) denotes all the nodes that are on the path from v_i to the root. If the user GID is revoked on time T_i, GID, T_i will be added into the revocation list RL. For the current time T, the binary tree BT and the revocation list RL the $\mathsf{KUNode}(BT, RL, T)$ function is defined as follows:

$\mathsf{KUNode}(BT, RL, T)$
  $X, Y \leftarrow \emptyset$
  $\forall (v_i, T_i) \in RL$
    if $T_i \leq T$ then add Path(v_i) to X
  $\forall x \in X$
    if $x_L \notin X$, then add x_L to Y
    if $x_R \notin X$, then add x_R to Y
  if $Y = \emptyset$, then add root to Y
  Return Y

To take an example, if there are four users $\theta_4, \theta_5, \theta_6, \theta_7$ issued by the authority θ_1, the revocation list $RL = \emptyset$ and the user θ_4 in Figure 1 is revoked currently, θ_4 would be added into $RL = \{\theta_4\}$. The result of $\mathsf{KUNode}(BT, RL, T)$ would be $Y = \{\theta_5, \theta_3\}$.

III. CCA2-PV-R-LU-MA-ABE

A. SYSTEM MODEL

There are five roles in this system, named the Cloud, Fogs, authorities, data owners and data visitors as described in Figure 2. The Cloud can complete traditional cloud computing missions, and there are many Fogs at the edge of the Cloud. Each Fog can manage lightweight computing missions for its local end nodes, including all kinds of sensors and devices. The Fogs are treated as a semi-trusted server in this scheme, which means the Fogs cannot get the data plaintext, but they should store the ciphertext and verify its validity, and re-encrypt the related ciphertext when the authorities update keys to data visitors. When a sensor collects and uploads data to the Fog, we treat the sensor as a data owner. And when a device runs some application that requests data from the Fog, we treat the device as a data visitor.

The Cloud sets up this system and declares system parameters. And in each Fog some authorities set up to generate attribute-related keys for nodes. The data visitor can query for secret keys to the authorities that is corresponding to their attributes. Before the data owners upload data to the Fog, they would decide the attributes that the data visitors should have to get access to such data, and they encrypt the data by the required access policy. When the Fog gets the ciphertext, it can verify the validation of the ciphertext without decryption or knowing any secret key, and only if the ciphertext is encrypted by declared attributes and not tampered, it can be transmitted in this Fog. If the data visitor's attributes satisfy the data's access policy, they can decrypt the ciphertext and get the data. Besides, if some data visitor's attribute is revoked, the secret key corresponding to this

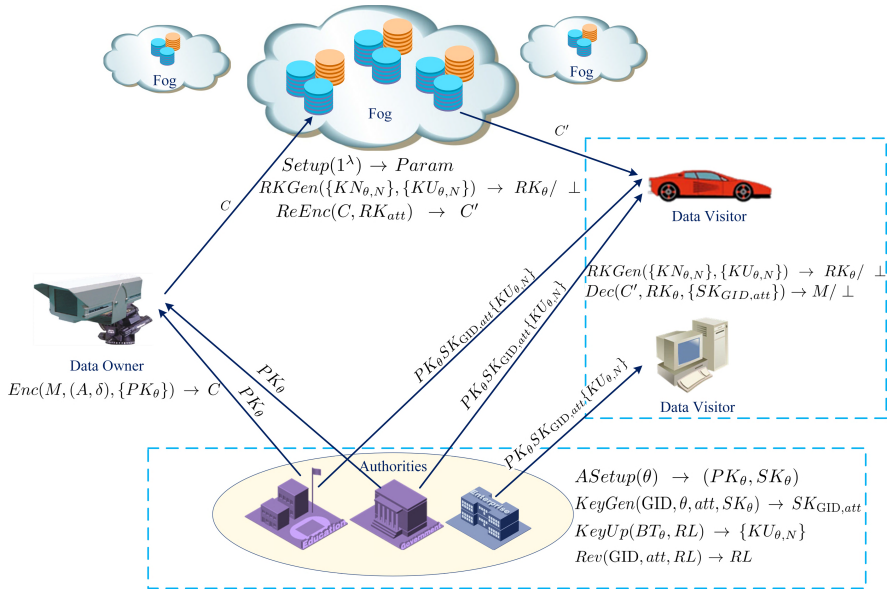

FIGURE 2. System model.

attribute will be revoked by the authority, and the Fog would get the revocation list and re-encrypt the ciphertext to prevent revoked users from decrypting it. To realize aforementioned functionality, this system consists of nine polynomial time algorithms defined as follows.

- $Setup(1^{\lambda}) \rightarrow Param$: This algorithm is run by the Cloud to set up the system. It takes security parameter as input and outputs the system parameters for Fogs.
- $ASetup(\theta) \rightarrow (PK_{\theta}, SK_{\theta})$: This algorithm is run by decentralized authorities to set up authority's public key and private key. It takes the authority's identity as input and then the authority publishes the public key and keeps the private key secretly.
- $KeyGen(\mathrm{GID}, \theta, att, SK_{\theta}) \rightarrow SK_{\mathrm{GID},att}$:This algorithm is run by authority to generate private keys. It takes the data visitor's identity GID, its attribute att and the authority's private key as input, and outputs the data visitor's private key.
- $KeyUp(BT_{\theta}, RL) \rightarrow \{KU_{\theta,N}\}$: This algorithm is run by each authority to update private keys to all attribute-unrevoked data visitors. It takes the revocation list and a binary tree as input, and outputs the updated re-encryption part of private keys. The updated part is published to the unrevoked data visitors and the Fog.
- $RKGen(\{KN_{\theta,N}\},\{KU_{\theta,N}\}) \rightarrow RK_{\theta}/\perp$: This algorithm is run by the Fog and unrevoked data visitors to get the updated re-encryption keys. It takes $\{KU_{\theta,N}\}$ and $\{KN_{\theta,N}\}$ as input and outputs the updated re-encryption keys.
- $Enc(M,(A,\delta),\{PK_{\theta}\}) \rightarrow C$: This algorithm is run by data owner to encrypt message with corresponding access policy. It takes the message with its access policy and the public key of related authorities as input, and outputs the ciphertext.
- $Verify(C,(A,\delta)) \rightarrow True/False$: The algorithm can be run publicly to verify the validation of ciphertext, and normally it is run by the Fog. It takes in the ciphertext and the access policy, and outputs $True$ if the ciphertext is valid.
- $ReEnc(C, RK_{att}) \rightarrow C'$: This algorithm is run by the Fog to re-encrypt the ciphertext to prevent revoked users from decrypting it.
- $Dec(C', RK_{\theta}, \{SK_{GID,att}\}) \rightarrow M/\perp$: This algorithm is run by the data visitor to decrypt. If the data visitor does not satisfy the access policy or some of its corresponding attributes have been revoked, it cannot get the plaintext correctly.
- $Rev(\mathrm{GID}, att, RL) \rightarrow RL$: This algorithm is run by the authority to add the revoked user's identity GID with attribute att to the revocation list.

B. SECURITY MODEL

We define the security model for CCA2-PV-R-LU-MA-ABE, named indistinguishability against selective authority and access policy and statically chosen ciphertext attacks (IND-sAA-sCCA2). At the beginning of this security model, the adversary should claim the access policy that it will challenge in the challenge phase, and it should also claim the corrupt authorities. The challenge message must be encrypted by at least one attribute from honest authority, and attributes from some of these corrupt authorities can also be used to encrypt the challenge message, which means in real condition if only part of the encrypted attributes are from corrupted authorities, the ciphertext still cannot be attacked successfully. Let $\Pi = \{Setup, ASetup, KeyGen, KeyUp, RKGen, Enc, Verify, ReEnc, Dec\}$ denote this CCA2-PV-R-LU-MA-ABE system.

Formally, we define the IND-sAA-sCCA2 security model for fog computing through the game played between adversary $\mathcal{A}$ and challenger $\mathcal{B}$.

Init. Adversary $\mathcal{A}$ should select a challenge access policy (A^*, δ^*) and a set of corrupt authorities $\mathcal{C}_\theta$ at the beginning of the game. Then $\mathcal{A}$ sends them to $\mathcal{B}$.

Setup. $\mathcal{B}$ sets up the system and sends public parameter *Param* to $\mathcal{A}$.

*Phase*1. $\mathcal{A}$ issues following queries statically:

- **ASetup**$(\cdot,\cdot)$ **Queries**: $\mathcal{A}$ outputs a set of non-corrupted authorities and $\mathcal{B}$ sets up the queried authorities.
- **KeyGen**$(\cdot,\cdot,\cdot,\cdot)$ **Queries**: $\mathcal{A}$ makes secret key queries for users with a set of its all attributes, and $\mathcal{B}$ generates secret keys to $\mathcal{A}$. If the queried attributes set satisfies the selected challenge access policy, we require $\mathcal{A}$ has to revoke one of these attributes from all users.
- **Rev**$(\cdot,\cdot,\cdot)$ **Queries**: $\mathcal{A}$ queries to revoke some users with certain attribute. $\mathcal{B}$ adds the identity and attribute into the revocation list, and updates keys to all unrevoked users, then we require Re-Encryption should be queried to re-encrypt the ciphertext with the latest re-encryption keys.
- **ReEnc**$(\cdot,\cdot)$ **Queries**: $\mathcal{A}$ makes re-encryption queries, then $\mathcal{B}$ runs the *RKGen* algorithm to get the latest re-encryption keys and takes it as input to run the *ReEnc* algorithm to get the re-encrypted ciphertext.
- **Dec**$(\cdot,\cdot,\cdot)$ **Queries**: $\mathcal{A}$ makes decryption queries for ciphertext with access policy. Then $\mathcal{B}$ generates private keys for such access policy and decrypts to respond to $\mathcal{A}$.

Challenge. $\mathcal{A}$ submits to $\mathcal{B}$ two messages M_0, M_1 with equal length. Then $\mathcal{B}$ randomly flips coin $b \in \{0,1\}$ and encrypts M_b with the selected access policy (A^*, δ^*). The ciphertext should be re-encrypted by the latest re-encryption keys, and then be returned to $\mathcal{A}$.

*Phase*2. It is the same as in *Phase*1 except that $\mathcal{A}$ cannot make KeyGen$(\cdot,\cdot,\cdot,\cdot)$ for the selected access policy or Dec$(\cdot,\cdot,\cdot)$ query for the challenge ciphertext.

Guess. Finally, $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$. $\mathcal{A}$ wins in the game if $b' = b$.

Definition 1: The CCA2-PV-R-LU-MA-ABE is indistinguishable against selective authority and access policy and statically chosen ciphertext attacks if for any probabilistic polynomial time adversary $\mathcal{A}$, the advantage of breaking the security game defined above is at most a negligible function.

IV. CONCRETE SCHEME

Based on the system model and security model in section III, we construct a concrete scheme on prime-order bilinear groups that perform more efficiently than composite-order bilinear groups. The proposed scheme is as follows.

- $Setup(1^\lambda) \rightarrow Param$: This algorithm is run by the Cloud. It takes as input the system security parameter 1^λ. Firstly, it picks a prime order bilinear group generator $\mathcal{G}$, and runs $(\mathbb{G}, p, g) \xleftarrow{R} \mathcal{G}(1^\lambda)$. Then it picks random exponents $\alpha_0, \beta_0 \xleftarrow{R} \mathbb{Z}_p$ as the master key, and computes $PK_0 = (e(g,g)^{\alpha_0}, g^{\beta_0})$ as the public key. For the identity and attribute are usually in a form of strings, it chooses a hash function H mapping identity GID to elements of $\mathbb{G}$, and another hash function F mapping attributes to elements of $\mathbb{G}$. Then it defines the attribute universe $\mathcal{U}$ and the authority universe $\mathcal{U}_\theta$, and $T : \mathcal{U} \rightarrow \mathcal{U}_\theta$ is a publicly computable function mapping each attribute to a unique authority, which means that, for example, T maps the attribute "$att_i@\theta_i$" to its relative authority "θ_i". It initializes a revocation list $RL = \emptyset$. Finally, it picks a Chameleon hash function $H_{ch} : \{0,1\}^* \rightarrow \mathcal{U}$ and runs the $KeyGen_{ch}(1^\lambda) \rightarrow (SK_{ch}, PK_{ch})$. The global parameter is $Param = (p, \mathbb{G}, g, H, F, T, H_{ch}, \mathcal{U}, \mathcal{U}_\theta, PK_0, PK_{ch}, RL)$. The system global parameter *Param* is the input for all of the following algorithms and we omit it for simplicity.
- $ASetup(\theta) \rightarrow (PK_\theta, SK_\theta)$: This algorithm is run by the authority θ. It takes as input the global parameters *Param* and the index of the authority $\theta \in \mathcal{U}_\theta \setminus \{0\}$. It picks two random exponents $\alpha_\theta, \beta_\theta \xleftarrow{R} \mathbb{Z}_p$, then publishes the public key $PK_\theta = (e(g,g)^{\alpha_\theta}, g^{\beta_\theta})$, and keeps the private key $SK_\theta = (\alpha_\theta, \beta_\theta)$.
- $KeyGen(\mathrm{GID}, \theta, att, SK_\theta) \rightarrow SK_{\mathrm{GID},att}$: This algorithm is run by the authority θ. It takes authority's private key SK_θ, user's identity GID and user's attribute $att \in \mathcal{U}$ as input and outputs the user's private key $SK_{\mathrm{GID},att}$. Firstly, it sets up a binary tree BT_θ if BT_θ has not been setup, in which each user with attribute $T(att) = \theta$ is assigned as a leaf node and it increases the height with users increasing. For each node $N \in BT_\theta$, it randomly picks $\{KN_{\theta,N}\} \xleftarrow{R} \mathbb{Z}_p$. Then authority θ randomly picks $r \xleftarrow{R} \mathbb{Z}_p$ and computes $K_{\mathrm{GID},att} = g^{\alpha_\theta} H(\mathrm{GID})^{\beta_\theta} F(att)^r$, $K'_{\mathrm{GID},att} = g^r$. Finally, it outputs $SK_{\mathrm{GID},att} = (K_{\mathrm{GID},att}, K'_{\mathrm{GID},att}, \{KN_{\theta,N}\}_{N \in Path(\mathrm{GID})})$, and gives $SK_{\mathrm{GID},att}$ to the user GID, then gives to the Fog $(BT_\theta, \{KN_{\theta,N}\}_{N \in Path(\mathrm{GID})})$.
- $KeyUp(BT_\theta, RL) \rightarrow \{KU_{\theta,N}\}$: This algorithm is run by the authority θ. It takes as input the revocation list RL and the binary tree BT_θ. Firstly, it randomly picks $RK_\theta \xleftarrow{R} \mathbb{Z}_p$ for BT_θ. Then it publishes $\{KU_{\theta,N}\} = \{RK_\theta \times KN_{\theta,N}\}_{N \in KUNode(BT_\theta, RL)}$.
- $RKGen(\{KN_{\theta,N}\}, \{KU_{\theta,N}\}) \rightarrow RK_\theta / \perp$: This algorithm is run by either the user GID or the Fog to generate the re-encryption key RK_θ for all unrevoked user with attribute $T(att) = \theta$. If $Path(\mathrm{GID}) \cap KUNode(BT_\theta, RL) = \emptyset$, it returns $\perp$. Otherwise, it calculates $N \in Path(\mathrm{GID}) \cap KUNode(BT_\theta, RL)$ and then it computes the re-encryption key $RK_\theta = \frac{KU_{\theta,N}}{KN_{\theta,N}}$.
- $Enc(M, (A, \delta), \{PK_\theta\}) \rightarrow C$: This algorithm is run by the data owner. It takes as input a message M, an access policy (A, δ) with $A \in \mathbb{Z}_p^{\ell \times n}$, and the public keys $\{PK_\theta\}$ of the relative authorities, where $\theta = T(att)$. We defines a function $\rho : [\ell] \rightarrow \mathcal{U}_\theta$ as $\rho(\cdot) = T(\delta(\cdot))$ to map rows to authorities. Then it randomly picks

$z, v_2, \ldots, v_n, \omega_2, \ldots, \omega_n \xleftarrow{R} \mathbb{Z}_p$ and creates vectors $\overrightarrow{v} = (z, v_2, \ldots, v_n)^\top$ and $\overrightarrow{\omega} = (0, \omega_2, \ldots, \omega_n)^\top$. We let $\lambda_x = \overrightarrow{A_x} \cdot \overrightarrow{v}$ denote the share of z corresponding to row x, and $\omega_x = \overrightarrow{A_x} \cdot \overrightarrow{\omega}$ denote the share of 0, where $\overrightarrow{A_x}$ is the x-th row of matrix A. Let $\omega_0 = \sum_{x\in\ell} \omega_x$. For each row x of A, it randomly picks exponent $t_x \xleftarrow{R} \mathbb{Z}_p$ and $t_0 \xleftarrow{R} \mathbb{Z}_p$, and computes the ciphertext $C = (C_0, \{C_{1,x}, C_{2,x}, C_{3,x}, C_{4,x}\}_{x\in[0,\ell]}, r_{ch})$ as follows:

$$
\begin{aligned}
C_0 &= M \cdot e(g,g)^z, C_{1,0} = e(g,g)^z e(g,g)^{\alpha_0 t_0},\\
C_{2,0} &= g^{-t_0}, C_{3,0} = g^{\beta_0 t_0} g^{\omega_0},\\
V &= H_{ch}(PK_{ch}, PK_{ch} \parallel C \setminus C_{4.0}, r_{ch}), C_{4,0} = F(V)^{t_0},\\
\{C_{1,x} &= e(g,g)^{\lambda_x + \alpha_{\rho(x)} t_x}, C_{2,x} = g^{-t_x},\\
C_{3,x} &= g^{\beta_{\rho(x)} t_x + \omega_x}, C_{4,x} = F(\delta(x))^{t_x}\}_{x\in[\ell]}
\end{aligned}
$$

- $Verify(C, (A, \delta)) \rightarrow True/False$: This algorithm is run by all roles in this system. It firstly computes $V' = H_{ch}(PK_{ch}, PK_{ch} \parallel C \setminus C_{4.0}, r_{ch})$ and verifies the validity of the ciphertext by computing:

$$e(C_{4,x}, g) \cdot e(C_{2,x}, F(att)) \stackrel{?}{=} 1 \quad (1)$$

$$e(C_{4,0}, g) \cdot e(C_{2,0}, F(V')) \stackrel{?}{=} 1 \quad (2)$$

$$\frac{\prod_{x\in[\ell]} (e(C_{2,x}, g^{\beta_{\rho(x)}}) \cdot e(C_{3,x}, g))}{e(C_{2,0}, g^{\beta_0}) \cdot e(C_{3,0}, g)} \stackrel{?}{=} 1 \quad (3)$$

  If all of these equations hold, it means that the ciphertext is encrypted exactly by the attribute att and not tampered maliciously, and then it outputs $True$. Else it outputs $False$.
- $ReEnc(C, RK_{att}) \rightarrow C'$: This algorithm is run by the Fog. The Fog re-encrypts the ciphertext by computing $C' = (C_0, C_{1,0}, C_{2,0}, C_{3,0}, C_{4,0}, \{C_{1,x}, C_{2,x}, C_{3,x}, C_{4,x}^{RK_{T(\delta(x))}}\}_{x\in[\ell]}, r_{ch})$. Then the re-encrypted ciphertext C' would be stored securely on the Fog.
- $Dec(C', RK_\theta, \{SK_{GID,att}\}) \rightarrow M/\perp$: This algorithm is run by the data visitor GID with attributes att. It firstly computes the original ciphertext $C = C'$ except that $\{C_{4,x}\}_{x\in[\ell]} = \{C'^{RK^{-1}_{T(\delta(x))}}_{4,x}\}_{x\in[\ell]}$. Then it verifies the validity of the ciphertext with the Equation (1), Equation (2) and Equation (3) in the algorithm $Verify(C, (A, \delta))$. Let (A, δ) be the access policy of the ciphertext. If the algorithm $Verify(C, (A, \delta))$ outputs $True$, it computes:

$$
\begin{aligned}
&C_{1,x} \cdot e(K_{\mathrm{GID},\delta(x)}, C_{2,x}) \cdot e(H(\mathrm{GID}), C_{3,x})\\
&\quad \cdot e(K'_{\mathrm{GID},\delta(x)}, C_{4,x})\\
&= e(g,g)^{\lambda_x} \cdot e(H(\mathrm{GID}), g)^{\omega_x}
\end{aligned}
$$

  Then it calculates a constants $c_x \in \mathbb{Z}_p$ such that $\sum_x c_x \overrightarrow{A_x} = (1, 0, \cdots, 0)$. For $\lambda_x = \overrightarrow{A_x} \cdot \overrightarrow{v}$ and $\omega_x = \overrightarrow{A_x} \cdot \overrightarrow{\omega}$, $(1, 0, \cdots, 0) \cdot \overrightarrow{v} = z$ and $(1, 0, \cdots, 0) \cdot \overrightarrow{\omega} = 0$. Finally, the data visitor computes:

$$M = \frac{C_0}{\prod_x (e(g,g)^{\lambda_x} e(H(\mathrm{GID}), g)^{\omega_x})^{c_x}} = \frac{M \cdot e(g,g)^z}{e(g,g)^z}$$

TABLE 2. Theoretical performance comparisons.

	LU-MA-ABE in [4]	This CCA2-PV-R-LU-MA-ABE
Security	Statically-secure	IND-sAA-sCCA2 secure
$SK_{\mathrm{GID},\lvert S\rvert}$ Size	$2k$	$(2+d)k$
$C_{(A,\rho)}$ Size	$4\ell + \lvert(A,\rho)\rvert + 1$	$4\ell + \lvert(A,\rho)\rvert + 5 + \lvert r_{ch}\rvert$
KeyGen Time	$(4\tau_e + 2\tau_m) \cdot \lvert S\rvert$	$(4\tau_e + 2\tau_m) \cdot \lvert S\rvert$
Encryption Time	$(6\ell+1)\tau_e + (2\ell+1)\tau_m$	$(6\ell+7)\tau_e + (2\ell+3)\tau_m + \tau_h$
Decryption Time	$(3\tau_p + 4\tau_m + \tau_e) \cdot \ell$	$(3\tau_p + 4\tau_m + 2\tau_e) \cdot \ell$

- $Rev(\mathrm{GID}, att, RL) \rightarrow RL$: This algorithm is run by each authority to revoke a certain attribute att from a user GID. It just adds this user's identity and revoked attribute into the revocation list RL.

V. SECURITY ANALYSIS

The proposed CCA2-PV-R-LU-MA-ABE concrete scheme is constructed based on the conference paper [5], and we add public-verifiable function to it. In this scheme, the executor runs $Verify(C, (A, \delta))$ algorithm with publicly claimed ciphertext and attributes, which has no contribution to breaking the security of the scheme. Thus the security proof is almost the same as the earlier version, and we omit it due to the limit of space.

VI. PERFORMANCE ANALYSIS

In Fog Computing, efficiency is one of the most important consideration when we design security mechanism. For the scheme is constructed based on prime-order bilinear groups and the revocation is run on the Subset-Cover Revocation Framework, the efficiency should be impressive. However, to enhance the security from CPA to CCA2, we apply the Chameleon Hash function and use this hash value as an attribute to encrypt a part of ciphertext, which brings some sacrifice of efficiency. Thus we analyze the efficiency of our scheme, and compare this scheme with Rouselakis-Waters statically-secure LU-MA-ABE scheme. In order to make the contrast more comprehensive, we compare both of the schemes in theoretical method and in experimental method.

A. THEORETICAL ANALYSIS

We compare the performance of our CCA2-PV-R-LU-MA-ABE scheme with Rouselakis-Waters statically-secure LU-MA-ABE scheme in Table 2. As shown in Table 2, for prime-order groups $\mathbb{G}$ and $\mathbb{G}_T$, we denote τ_e as one exponent operation time, τ_m as one multiplication operation, τ_p as one pairing operation time, and τ_h as one Chameleon hash operation time. In this table, the secret key $SK_{\mathrm{GID},|S|}$ is associated with the attribute set S with $k = |S|$ and the node set $Path(\mathrm{GID})$ with $d = |Path(\mathrm{GID})|$. The ciphertext $C_{(A,\rho)}$ is associated with the LSSS-policy (A, ρ) with $A \in \mathbb{Z}_p^{\ell\times n}$.

In the comparisons, the extra length of secret key comes from key revocation function, which is equal to $d \times k$ where d is logarithm of the total number of users in the authority. The extra length of ciphertext comes from the part encrypted by Chameleon hash value, which is equal to $4 + |r_{ch}|$, and the size can be seen as insignificant. The time of key generation is the same in both scheme. The extra time of encryption comes

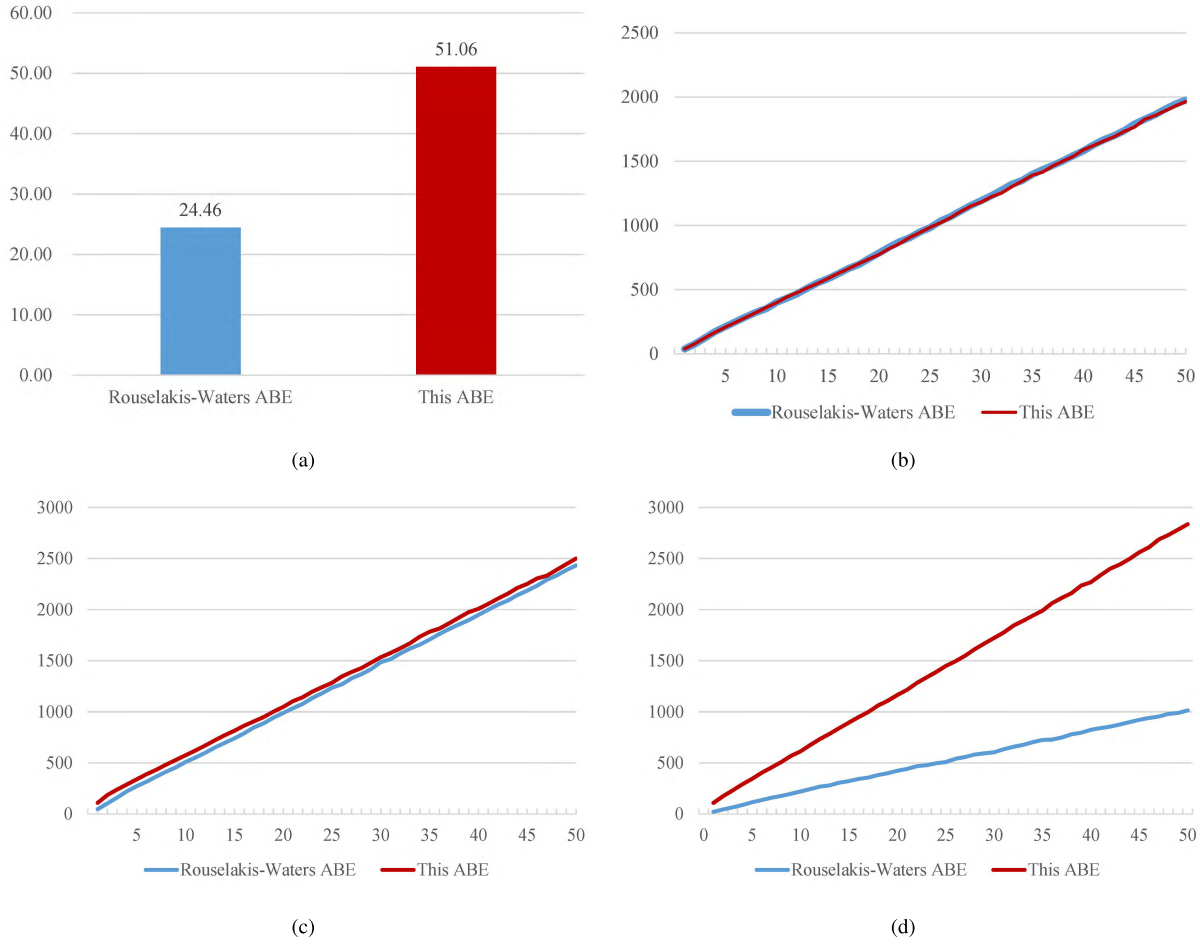

FIGURE 3. Experimental performance comparisons. (a) Setup times (ms). (b) KeyGen times (ms). (c) Encryption times (ms). (d) Decryption times (the red line includes verification times) (ms).

from the execution of Chameleon hash of ciphertext and a part of ciphertext encrypted by this Chameleon hash value. The extra time of decryption in our scheme is only one exponential operation because of the re-encrypted ciphertext should be decrypted by the update key from key revocation function. Through the overall theoretical comparison, the additional overhead for CCA2 security and revocation function is considerable low.

B. EXPERIMENTAL ANALYSIS

We implement our proposed CCA2-PV-R-LU-MA-ABE scheme and the Rouselakis-Waters Statically-secure LU-MA-ABE with the Java Pairing-Based Cryptography Library (JPBC) [44] to evaluate and compare their performances. The experiment is run on 3.7GHz Inter Xeon E5-1620 v2 platform with 16GB RAM running 64-bit Windows 10 Professional operation system. The elliptic curve is Type A with $y^2 = x^3 + x$ for Tate symmetric pairings, and the primes are 512-bit large randomly picked by the system. We run both of the schemes for 100 rounds and calculate the average execution time they take, where the number of attributes is ranging from 1 to 50. Finally, we draw the graphs in figure 3.

We separately demonstrate *Setup* and *ASetup* time in Figure 3a, *KeyGen* time in Figure 3b, *Encryption* time in Figure 3c, and *Decryption* time in Figure 3d. It can be seen that the performance of our IND-sAA-sCCA2 secure R-LU-MA-ABE scheme is almost the same as the performance of Rouselakis-Waters Statically-secure LU-MA-ABE. In Figure 3a, the setup time difference between both schemes mainly comes from initialization of Chameleon hash function. In Figure 3b, our scheme only picks more random elements than the Rouselakis-Waters ABE, and thus the curves almost coincide. In Figure 3c, our scheme encrypts one part of ciphertext, and thus the time difference is almost fixed, while with the increase of the attribute number, the proportion of time difference decreases. Figure 3d displays *Decryption* time, where the verification time is also included, and if the public verification process is executed by the Fogs, the *Decryption* time of the both schemes is almost the same too. Thus the experimental verification shows that the additional overhead for CCA2 security is considerable low.

VII. FUTURE IMPLEMENTATION ON BLOCKCHAIN

The proposed system allows decentralized authorization and public verification, and Blockchain has some same respects

with Fog Computing. Thus this system can also be utilized in blockchain to support flexible data access control, especially in blockchain-based distributed storage system. Files can be encrypted with this system before updating to blockchain, and the file owner should declare the access attributes. Miners and nodes in blockchain can verify the validity of the updated files with the public verification algorithm, and only the valid files can be stored. Data visitor can generate its private keys from different distributed authorities based on the attributes. And the data visitor can search the declared attributes of all files and download the attribute-matching files that it can access to. This system is CCA2 secure to ensure security of the files, and it can be executed efficiently.

VIII. CONCLUSION

An efficient CCA2 secure publicly-verifiable revocable large-universe multi-authority CP-ABE system for Fog Computing was defined in this paper, and we constructed a concrete scheme based on prime-order bilinear groups, which has superior performance than composite-order bilinear groups. Then we proved this scheme is IND-sAA-sCCA2 secure under the (q+1)-DPBDHE2 assumption. We analyzed the performance of this scheme with both theoretical calculation and experimental verification methods, and this scheme performs a remarkable efficiency. Thus the proposed scheme is secure and efficient enough to serve the practical Fog Computing. This system achieves flexible and publicly-verifiable access control, and it will have a wide range of application scenarios in the future.

REFERENCES

[1] Z. Zhou, M. Dong, K. Ota, G. Wang, and L. T. Yang, "Energy-efficient resource allocation for D2D communications underlaying cloud-RAN-based LTE-A networks," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 428–438, Jun. 2016.

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 3494, R. Cramer, Ed. Aarhus, Denmark: Springer, 2005, pp. 457–473.

[3] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography* (Lecture Notes in Computer Science), vol. 4392, S. P. Vadhan, Ed. Amsterdam, The Netherlands: Springer, 2007, pp. 515–534.

[4] Y. Rouselakis and B. Waters, "Efficient statically-secure large-universe multi-authority attribute-based encryption," in *Financial Cryptography and Data Security* (Lecture Notes in Computer Science), vol. 8975, R. Böhme and T. Okamoto, Eds. San Juan, Puerto Rico: Springer, 2015, pp. 315–332.

[5] D. Li, J. Chen, J. Liu, Q. Wu, and W. Liu, "Efficient CCA2 secure revocable multi-authority large-universe attribute-based encryption," in *Proc. 9th Int. Symp. Cyberspace Saf. Secur. (CSS)*, Xi'an, China, Oct. 2017, pp. 103–118.

[6] S. Wen, W. Wu, and A. Castiglione, Eds., *Cyberspace Safety and Security: 9th International Symposium, CSS 2017, XiâĂŹan China, October 23–25, 2017, Proceedings*, vol. 10581. Xi'an, China: Springer, 2017.

[7] F. Bonomi, R. A. 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., (MCC@SIGCOMM)*, Helsinki, Finland, M. Gerla and D. Huang, Eds. Helsinki, Finland: ACM, 2012, pp. 13–16.

[8] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Workshop Mobile Big Data*, 2015, pp. 37–42.

[9] V. Pande, C. Marlecha, and S. Kayte, "A review-fog computing and its role in the Internet of Things," *Int. J. Eng. Res. Appl.*, vol. 6, no. 10, pp. 1–5, 2016.

[10] Y. Zhou, D. Zhang, and N. Xiong, "Post-cloud computing paradigms: A survey and comparison," *Tsinghua Sci. Technol.*, vol. 22, no. 6, pp. 714–732, 2017.

[11] A. Rayes and S. Salam, "Fog computing," in *Internet of Things From Hype to Reality*. Springer, 2019, pp. 155–180.

[12] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the Internet of Things," in *Proc. 2nd ACM SIGCOMM Workshop Mobile Cloud Comput. (MCC@SIGCOMM)*, Hong Kong, M. Gerla and D. Huang, Eds. Hong Kong, China: ACM, 2013, pp. 15–20.

[13] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Warsaw, Poland, Sep. 2014, pp. 1–8.

[14] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurrency Comput., Pract. Exper.*, vol. 28, no. 10, pp. 2991–3005, 2016.

[15] Y. Jiang, W. Susilo, Y. Mu, and F. Guo, "Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 720–729, Jan. 2017.

[16] 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 Gener. Comput. Syst.*, vol. 78, pp. 753–762, Jan. 2018.

[17] 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)*, Alexandria, VA, USA, 2006, pp. 89–98.

[18] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (S&P)*, Oakland, CA, USA, May 2007, pp. 321–334.

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

[20] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, Alexandria, VA, USA, Oct. 2007, pp. 195–203.

[21] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably secure and efficient bounded ciphertext policy attribute based encryption," in *Proc. ACM Symp. Inf., Comput. Commun. Secur. (ASIACCS)*, Sydney, NSW, Australia, 2009, pp. 343–352.

[22] 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)*, Taormina, Italy, in Lecture Notes in Computer Science, vol. 6571, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds. Taormina, Italy: Springer, 2011, pp. 53–70.

[23] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 6223, T. Rabin, Ed. Santa Barbara, CA, USA: Springer, 2010, pp. 191–208.

[24] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 6110, H. Gilbert, Ed. French Riviera: Springer, 2010, pp. 62–91.

[25] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, Chicago, IL, USA, 2009, pp. 121–130.

[26] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," in *Progress in Cryptology—INDOCRYPT* (Lecture Notes in Computer Science), vol. 5365, D. R. Chowdhury, V. Rijmen, and A. Das, Eds. Kharagpur, India: Springer, 2008, pp. 426–436.

[27] S. Müller, S. Katzenbeisser, and C. Eckert, "Distributed attribute-based encryption," in *Proc. 11th Int. Conf. Inf. Secur. Cryptol. (ICISC)*, Seoul, South Korea, in Lecture Notes in Computer Science, vol. 5461, P. J. Lee and J. H. Cheon, Eds. Seoul, South Korea: Springer, 2008, pp. 20–36.

[28] A. B. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 6632, K. G. Paterson, Ed. Tallinn, Estonia: Springer, 2011, pp. 568–588.

[29] 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. Symp. Res. Comput. Secur. (ESORICS)*, Leuven, Belgium, in Lecture Notes in Computer Science, vol. 6879, V. Atluri and C. Díaz, Eds. Leuven, Belgium: Springer, 2011, pp. 278–297.

[30] A. B. Lewko and B. Waters, "Unbounded HIBE and attribute-based encryption," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 6632, K. G. Paterson, Ed. Tallinn, Estonia: Springer, 2011, pp. 547–567.

[31] A. B. Lewko, "Tools for simulating features of composite order bilinear groups in the prime order setting," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 7237, D. Pointcheval and T. Johansson, Eds. Cambridge, U.K.: Springer, 2012, pp. 318–335.

[32] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Berlin, Germany, 2013, pp. 463–474.

[33] N. Attrapadung and H. Imai, "Attribute-based encryption supporting direct/indirect revocation modes," in *Proc. 12th IMA Int. Conf., Cryptogr. Coding*, in Lecture Notes in Computer Science, vol. 5921, M. G. Parker, Ed. Cirencester, U.K.: Springer, 2009, pp. 278–300.

[34] A. Beimel, "Secure schemes for secret sharing and key distribution," Ph.D. dissertation, Dept. Comput. Sci., Israel Inst. Technol., Haifa, Israel, 1996.

[35] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 7417, R. Safavi-Naini and R. Canetti, Eds. Santa Barbara, CA, USA: Springer, 2012, pp. 199–217.

[36] H. Tsuchida, T. Nishide, E. Okamoto, and K. Kim, "Revocable decentralized multi-authority functional encryption," in *Progress in Cryptology—INDOCRYPT* (Lecture Notes in Computer Science), vol. 10095, O. Dunkelman and S. K. Sanadhya, Eds. Kolkata, India, 2016, pp. 248–265.

[37] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 3027, C. Cachin and J. Camenisch, Eds. Interlaken, Switzerland: Springer, 2004, pp. 207–222.

[38] S. Yamada, N. Attrapadung, G. Hanaoka, and N. Kunihiro, "Generic constructions for chosen-ciphertext secure attribute based encryption," in *Public Key Cryptography—(PKC)* (Lecture Notes in Computer Science), vol. 6571, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds. Taormina, Italy: Springer, 2011, pp. 71–89.

[39] C. Chen, Z. Zhang, and D. Feng, "Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost," in *5th Int. Conf. Provable Secur. (ProvSec)*, Xi'an, China, in Lecture Notes in Computer Science, vol. 6980, X. Boyen and X. Chen, Eds. Xi'an, China: Springer, 2011, pp. 84–101.

[40] A. Ge, R. Zhang, C. Chen, C. Ma, and Z. Zhang, "Threshold ciphertext policy attribute-based encryption with constant size ciphertexts," in *Proc. 17th Australas. Conf. Inf. Secur. Privacy (ACISP)*, Wollongong, NSW, Australia, in Lecture Notes in Computer Science, vol. 7372, W. Susilo, Y. Mu, and J. Seberry, Eds. Wollongong, NSW, Australia: Springer, 2012, pp. 336–349.

[41] W. Liu, J. Liu, Q. Wu, B. Qin, and Y. Zhou, "Practical direct chosen ciphertext secure key-policy attribute-based encryption with public ciphertext test," in *Computer Security—(ESORICS)* (Lecture Notes in Computer Science), vol. 8713, M. Kutylowski and J. Vaidya, Eds. Wroclaw, Poland: Springer, 2014, pp. 91–108.

[42] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 2139, J. Kilian, Ed. Santa Barbara, CA, USA: Springer, 2001, pp. 41–62.

[43] D. Li, J. Liu, Z. Zhang, Q. Wu, and W. Liu, "Revocable hierarchical identity-based broadcast encryption," *Tsinghua Sci. Technol.*, vol. 23, no. 5, pp. 539–549, 2018.

[44] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proc. 16th IEEE Symp. Comput. Commun. (ISCC)*, Kerkyra, Greece, Jun./Jul. 2011, pp. 850–855.

[45] P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds., *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*. Alexandria, VA, USA: ACM, 2007.

[46] D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds., *Public Key Cryptography—(PKC) 2011 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6–9, 2011. Proceedings* (Lecture Notes in Computer Science), vol. 6571. Taormina, Italy: Springer, 2011.

[47] K. G. Paterson, Ed., *Advances in Cryptology—EUROCRYPT 2011: 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011, Proceedings* (Lecture Notes in Computer Science), vol. 6632. Tallinn, Estonia: Springer, 2011.

DAWEI LI was born in Shandong, China. He received the B.S. degree from Beihang University, Beijing, China, in 2015, where he is currently pursuing the Ph.D. degree in electronic and information engineering. His research interests include applied cryptography and blockchain.

JIANWEI LIU was born in Shandong, China. He received the B.S. and M.S. degrees in electronic and information from Shandong University, Shandong, in 1985 and 1988, respectively, and the Ph.D. degree in communication and electronic system from Xidian University, Shaanxi, China, in 1998. He is currently a Professor of cyber science and technology with Beihang University, Beijing, China. His current research interests include wireless communication networks, cryptography, and information and network security.

QIANHONG WU was born in Sichuan, China. He received the Ph.D. degree in cryptography from Xidian University, Shanxi, China, in 2004. Since 2004, he has been with Wollongong University, Australia, as an Associate Research Fellow, with Wuhan University, China, as an Associate Professor, and with Universitat Rovira i Virgili, Catalonia, as a Research Director. He is currently a Professor with Beihang University, China. His current research interests include cryptography, data security and privacy, and information theory. He is a member of the IACR, the ACM, and the IEEE.

ZHENYU GUAN received the Ph.D. degree in electronic engineering from Imperial College London, U.K., in 2013. Since 2013, he has been with Beihang University, China, as a Lecturer. His current research interests include cryptography engineering, security of IOT, and blockchain. He is a member of the IEEE and the IEICE.