This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2705076, IEEE Access

An Attribute-Based Encryption Scheme to Secure Fog Communications

Arwa Alrawais [*] [‡], Abdulrahman Alhothaily[*] [†], Chunqiang Hu [*] [¶], Xiaoshuang Xing[§], and Xiuzhen Cheng[*]
[*]Department of Computer Science, The George Washington University, Washington, DC 20052, USA
[‡]College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, KSA
[†]General Department of Payment Systems, Saudi Arabian Monetary Authority, Riyadh, KSA
[¶]The School of Software Engineering, Chongqing University, Chongqing, China
[§]School of Computer Science and Engineering, Changshu Institute of Technology, Changshu, China
Email: {alrawais,hothaily, cheng}@gwu.edu, hcq0394@gmail.com, xing@cslg.edu.cn
Corresponding authors: Xiaoshuang Xing and Chunqiang Hu

***Abstract*—Fog computing is deemed as a highly virtualized paradigm that can enable computing at the Internet of Things (IoT) devices residing in the edge of the network, for the purpose of delivering services and applications more efficiently and effectively. Since fog computing originates from and is a non-trivial extension of cloud computing, it inherits many security and privacy challenges of cloud computing, causing the extensive concerns in the research community. To enable authentic and confidential communications among a group of fog nodes, in this paper, we propose an efficient key exchange protocol based on Ciphertext-Policy Attribute Based Encryption (CP-ABE) to establish secure communications among the participants. To achieve confidentiality, authentication, verifiability, and access control, we combine CP-ABE and digital signature techniques. We analyze the efficiency of our protocol in terms of security and performance. We also implement our protocol and compare it with the certificate-based scheme to illustrate its feasibility.**

Keywords—Fog computing; Security; Ciphertext-Policy Attribute Based Encryption (CP-ABE); Cloud computing; Communications security.

I. Introduction

Fog computing is a promising computing paradigm that extends cloud computing to the edge of the network. It enables a new breed of applications and services such as location awareness, quality of services (QoS) enhancement, and low latency. Fog computing can provide these services with elastic resources at low cost. It also enables the smooth convergence between cloud computing and IoT devices for content delivery. As promising as it is, fog computing is facing many security issues. Secure communications are among the issues that raise the most concerns from users when they use fog computing to transmit their data to the cloud to be stored and processed. In general, the significant threats in fog computing networks are:

- Data alteration: An adversary can compromise data integrity by attempting to modify or destroy the legitimate data. Hence, it is essential to define a security mechanism to provide data integrity verification of the transmitted data between the fog nodes and the cloud.
- Unauthorized access: An adversary can gain accesses to unauthorized data without permission or qualifications, which could result in loss or theft of data. This attack raises a security issue that could expose a user's private information.
- Eavesdropping attacks: eavesdroppers can gain unauthorized interception to learn a lot about the user information transmitted via wireless communications. The risk of such attacks is that they cannot be easily detected because eavesdropping does not change anything in the network operations.

The primary security requirements for the communications between the fog nodes and the cloud are: confidentiality, access control, authentication, and verifiability. To effectively defend against the aforementioned threats, we need an efficient security mechanism that can satisfy the primary security requirements. Attribute-Based Encryption (ABE) developed by [1] is a promising solution that can provide some of the security requirements. ABE is a public key based on one-to-many encryption that employs the user's identity as an attribute. In ABE, a set of attributes and a private key computed from the attributes are respectively used for encryption and decryption. There are two main types of ABE systems: Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE). In KP-ABE [2], [3], the roles of the attributes are used to describe the ciphertext and an access policy is associated with the user's private key; while in CP-ABE [4], [5], [1], [6], the attributes are associated with the user's private key and the ciphertext is associated with an access policy. In this paper, we develop an encrypted key exchange protocol based on Ciphertext-Policy Attribute Based Encryption (CP-ABE) to enable authenticated and confidential communications between fog nodes and the cloud. The protocol establishes secure communications to exchange the shared key that can be used to encrypt and decrypt the exchanged information. Each fog node can obtain the shared key only if the fog node satisfies the policy defined over a set of attributes which is attached to the ciphertext.

Contribution: In this paper, we propose a novel encrypted key exchange protocol based on CP-ABE for secure communications in a fog computing network, which features the following achievements:

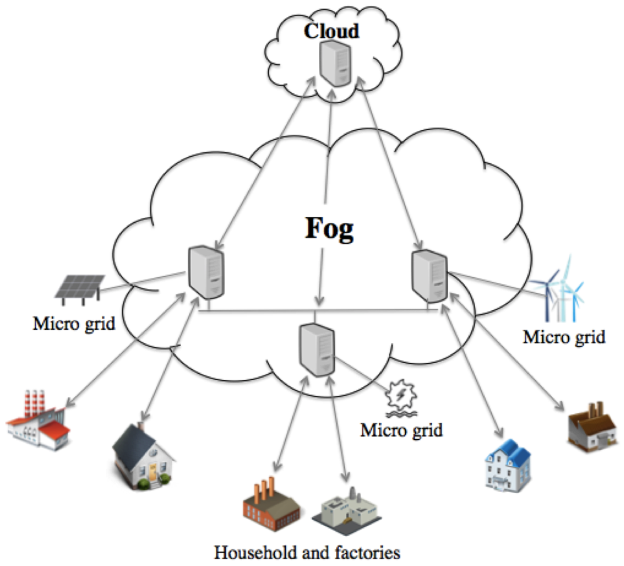


Fig. 1: An example smart grid based on fog computing.

- We develop a protocol for encrypted key exchange based on CP-ABE that combines encryption and signature to achieve a fine-grained data access control, confidentiality, authentication, and verifiability.
- We discuss the security of our protocol and prove its correctness. In particular, we investigate the security of our protocol under different attack scenarios.
- We analyze the performance of our proposed protocol and illustrate its efficiency in terms of message size and communication overhead.
- We implement and compare our protocol with a certificate-based protocol and shows its feasibility.

This paper is structured as follows. Section II presents the motivation of this study and sketches an overview of the related work. In Section III, we introduce the network model of our protocol. Section IV describes our proposed protocol. Section V and Section VI respectively present the security analysis and performance study of the protocol. In Section VII, we report the implementation results of our protocol. This paper is concluded in Section VIII.

II. Motivation and Related Work

A. Motivation

One of the real world applications that motivates our problem formulation is smart grids. A smart grid system is an electrical grid that intelligently controls, measures, and balances energy. It can automatically change to a different energy resource depending on the availability and the energy demand, which can help consumers optimize their consumption and lower the cost of the bill. A smart grid system consists of suppliers, cloud, and grid sensors or devices as shown in Fig. 1. Each smart grid gathers data and sends it back to the cloud via fog to analyze the behaviors of the consumers and the suppliers. Then, the smart grid acts based on the results of the analysis of the collected data. At this point, it is clear that the smart grid system requires real-time processing and distributed control. Fog computing can provide an interplay between real-time and batch analytics, but it also introduces new security challenges. In particular, attackers can easily launch many attacks when data is transmitted via a wireless channel and expose the users' information. Specifically, the transmitted data between fog nodes and the cloud for processing purposes allow the adversary to launch more sophisticated attacks. Additionally, existing protocols suffer from several drawbacks as mentioned in Section VI-C. Thereby, we need an efficient protocol to establish secure communications between fog nodes and the cloud.

B. Related Work

The main purpose of this paper is to propose an encrypted key exchange protocol based on CP-ABE to resist several sophisticated attacks in the fog computing network. Hence, in this section, we summarize the most closer works along two lines:

- ABE: Several existing researches [7], [8], [9], [10], [11] utilize ABE as a part of their proposed solution to achieve different security objectives. In [12], the authors proposed a patient-centric framework for data sharing access control to personal health record stored in cloud servers. They used the ABE techniques to achieve a high degree of the user's privacy and a fine-grained data access control for personal health records. Another effort in [13] combined KP-ABE with other techniques to simultaneously achieve data confidentiality and scalable data access control in the cloud server. Recently, the authors in [14] proposed a novel CP-ABE scheme for data-sharing to enforce an efficient data access control based on the data sharing characteristics.
- Fog computing: The fog computing platform provides a highly scalable solution for IoT devices and applications. Many works discussed the role of fog computing in IoT environment. In [15], the authors discussed the security and privacy challenges of fog computing in IoT environments. Fundamentally, they described how to use fog computing to enhance the security and privacy issues in IoT environments. Additionally, Kirak, *et. al.* [16] analyzed the programming model for large scale and latency sensitive IoT applications utilizing the fog computing platform. They studied the model with a camera network and connected vehicle applications and showed the efficient role of fog computing in IoT. Another work [17] evaluated the suitability of fog computing in the context of IoT environments. The authors developed a mathematical model to evaluate the applicability of fog computing and compared it with the traditional cloud computing in terms of latency, cost, and power consumption. Their results depicted the efficiency, provisioned QoS, and eco-friendliness of fog computing in IoT technology compared to cloud computing. Recent works have demonstrated the role of fog computing

on more specific IoT applications. Nguyen *et. al.* [18] proposed a Software Defined Network (SDN) based on vehicle ad hoc networking supported by fog computing. The proposed architecture solves many issues in vehicle ad hoc networks by increasing the connectivity between vehicles, vehicle-to-infrastructure, and vehicle-to-base-station while integrating fog computing to reduce latency and provide resource utility. The work in [19] introduced the fog platform as a novel solution for energy management. They illustrated the energy management as a service over fog computing on two different domains of home energy management and micro grid-level energy management. Their results showed that fog computing can improve efficiency, flexibility, interoperability, and connectivity, and can minimize the cost and time of energy management services. Another effort in [20] focused on health care applications, specifically a pervasive health monitoring application, which requires low latency and low network overhead. The authors employed fog computing to monitor falls or strokes by analyzing the data throughout the network and provide real-time detection. Their experiments showed that the proposed system achieves a low miss rate and low false positive rate.

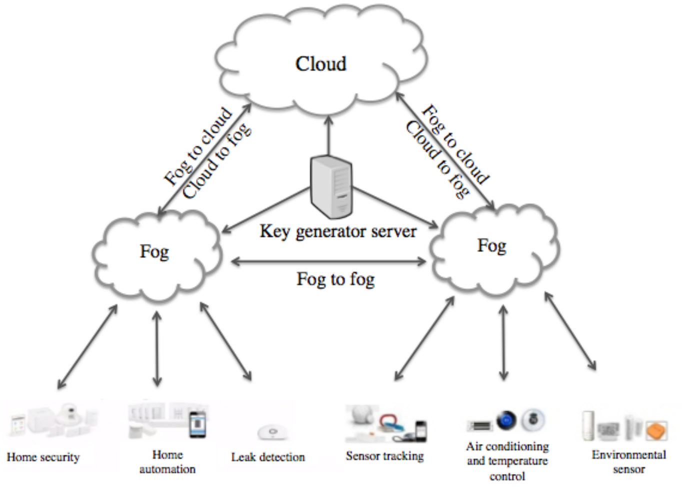


Fig. 2: Our proposed protocol.

III. Network Model and Preliminaries

1) Network Model: A representative network architecture for fog and cloud computing is illustrated in Fig. 2. This network architecture is composed of the following entities: a cloud, a key generator server, fog nodes, and IoT devices. The key generator server is used to generate and distribute the keys among the involved entities. The cloud defines the access structure $\mathbb{A}$ and performs the encryption to get ciphertext. We assume that the access structure $\mathbb{A}$ is given to all fog nodes. The fog node carries a set of attributes that is defined by an access structure $\mathbb{A}$ associated with the ciphertext. In particular, we assume that each fog node is associated with S attributes that can be viewed as a meaningful string of arbitrary length. For example, each fog node can have the following set $fog_i = \{model\_number, manufactured\_company, location\}, \ldots$ $fog_n = \{location, model\_number\}$. Thus, fog_i can execute the protocol to establish secure communications with other fog nodes and the cloud, if only its attributes set S_i satisfies $\mathbb{A}$. Thereby, a party of fog nodes whose attributes satisfy the access structure $\mathbb{A}$ can compute the shared key.

2) Preliminaries:

- Bilinear Pairing: Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two multiplicative cyclic groups of prime order p and g be a generator of $\mathbb{G}_1$. The bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ has the following properties:
  1) Bilinearity: For all $u, v \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
  2) Non-degeneracy: The generator g should satisfy $e(g, g) \neq 1$.
  3) Computable: For any $u, v \in \mathbb{G}_1$, there exists an efficient algorithm to compute $e(u, v)$.
- Definition 1 (**A**ccess Structure [21]). Let $\{P_1, \ldots P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,\ldots P_n\}}$ is monotone if for $\forall B, C : B \in \mathbb{A}$ and $C \subseteq B$, which implies $C \in \mathbb{A}$. A Monotone Access Structure (MAS) is a monotone collection $\mathbb{A}$ of non-empty subsets of $\{P_1, \ldots P_n\}$. The sets in $\mathbb{A}$ are called authorized sets, and the sets, which are not in $\mathbb{A}$, are called unauthorized sets.
  In our protocol, a set of attributes plays the role of an entity, and an access structure $\mathbb{A}$ specifies the policy of the set of attributes. As proposed by [1], we restrict our attention to monotonic access structures, which is referred to as an access structure throughout the paper.

3) Security Goals: Our main security goals are to establish secure communications in the fog computing network. Thus, the system should achieve the following security objectives:

- Confidentiality: Sensitive data should be only disclosed to legitimate entities. In our system, we utilize CP-ABE to ensure confidentiality of the transmitted data.
- Authentication: The system should prevent an active adversary who does not have the privilege to change or learn information of the transmitted data. Thus, a proper security mechanism should be adopted to ensure the authenticity of the data.
- Access Control: To reduce the risk of data exposure by an active adversary, a fine-grained access control should be enforced. The primary goal of our scheme design is to exchange the shared key securely; however, our scheme can be utilized to grant different access rights for each fog node in the same group.
- Verifiability: From the entity's signature, the fog node can be convinced that the message is generated by the same entity.

IV. Our Proposed Protocol

In order to achieve the security requirements of the communications between fog nodes and the cloud, we propose an encrypted key exchange protocol based on CP-ABE [22], [1].

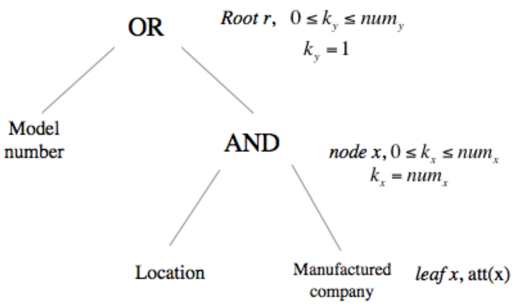


Fig. 3: An example of an access structure.

More specifically, we design a protocol such that each fog node is associated with a set of attributes, and assign each ciphertext with an expressive access structure that is defined over these attributes. This feature enforces the decryption procedure based on the fog node's attributes. Each ciphertext carries an access structure such that the fog can decrypt the ciphertext and obtain the shared key only if it possesses the specified attributes in the access structure. In this section, we propose our protocol based on the combination of CP-ABE and digital signature techniques. First, we define the access structure of our protocol. Then, we detail our protocol algorithms.

In our protocol, we utilize an access tree proposed by [1] as an access structure $\mathbb{A}$, which is shown in Fig. 3. Let T be a tree representing an access structure, where each non-leaf node is a threshold gate, and each leaf node describes an attribute. Assume that num_x is the number of children of node x, and k_x is the threshold value, then $0 \leqslant k_x \leqslant num_x$. Each interior node x is associated with two parameters k_x and num_x. The threshold value k_x outputs 1 if it is an OR gate, and outputs num_x if it is an AND gate. For each leaf node x, we define the threshold value to be $k_x = 1$. To facilitate the access tree structure description, the following functions are defined: $parent(x)$ is the parent of the node x in the tree, $att(x)$ is the attribute of the leaf node x, and $index(x)$ is the function that returns a uniquely assigned number that is associated with node x.

To satisfy the access tree, let T be a tree with a root node R, and let T_x be a subtree rooted at node x. If a set of the attributes γ satisfies T_x, then $T_x(\gamma) = 1$. We compute $T_x(\gamma) = 1$ recursively as follows:

- If x is a non-leaf node, we evaluate $T_x(\gamma)$ based on the children of x; if and only if at least k_x of the children return 1, $T_x(\gamma) = 1$.
- If x is a leaf node, then $T_x(\gamma) = 1$ if and only if $att(x) \in \gamma$.

At the beginning of our protocol, each fog node is associated with an access structure $\mathbb{A}$. The protocol can be executed with the following algorithms: Setup, Key Generation, Encryption, and Decryption. A private key is issued for each fog node based on the corresponding attribute set S. Then, the cloud runs the encryption algorithm that outputs an encrypted symmetric key. The cloud broadcasts the encrypted key to a group of fog nodes. Upon receiving the encrypted key, each fog node runs the decryption algorithm using its private key to extract the symmetric key. Our protocol consists of four algorithms that are detailed as follows:

Algorithm 1 describes the system setup and is executed by the key generator server. It takes the security parameter K as an input, publishes the public parameters PK to all involved entities, and holds the master key MK.

Algorithm 1 Setup (K)

```
Choose bilinear groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $p$; and the generators $g_1$ and $g_2$;
Choose three random exponents $\alpha, \beta_1, \beta_2 \in \mathbb{Z}_p$ such that $\beta_1 \neq \beta_2 \neq 0$;
Select a hash function as a random oracle $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$;
The public key is published as:
    $PK = (\mathbb{G}_1, \mathbb{G}_2, g_1, g_2, p, H, h_1 = g_1^{\beta_1}, h_2 = g_1^{\beta_2}, e(g_1, g_1)^{\alpha})$   (1)
The master key is $MK = (\beta_1, \beta_2, g_1^{\alpha})$
```

Algorithm 2 is also performed by the key generator server to generate the secret key SK that belongs to an entity specified by its set of attributes S. It takes the public parameters PK, the master key MK, and the set of attributes S to generate the secret key SK for the entity possessing S.

Algorithm 2 Key Generation (MK, PK, S)

```
Generate a key pair $(s_k, v_k)$ and select randoms $r, r_v \in \mathbb{Z}_p$;
Broadcast $v_k$ to others to verify the entity that belongs to $S$;
for Each $j \in S$ do
    Choose $r_j \in \mathbb{Z}_p$ and compute
    $D_j = g_1^r \cdot H(j)^{r_j}$ and $D'_j = g_1^{r_j}$
end for
The secret key $SK$ belonging to $S$ is computed as:
    $SK = (D = g_1^{\alpha + r/\beta_1}, E = g_1^{r/\beta_2}, \forall j \in S : D_j, D'_j)$   (2)
```

Algorithm 3 provides the details of the encrypted shared key K. It is executed by the cloud that takes as inputs the public parameters PK and the access tree structure T. It outputs the ciphertext C that contains the symmetric key.

Algorithm 4 describes the decryption procedure to obtain a shared symmetric key. This algorithm is executed by each fog node, which takes as inputs the public parameters PK, the secret key SK, and the ciphertext C. Then, it outputs either the symmetric key K or $\perp$. Note, the Lagrange's coefficient $\triangle_{i,S}$ for $i \in \mathbb{Z}_p$ and a set of elements in $\mathbb{Z}_p$ is defined as $\triangle_{i,S} = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$. Note that **Algorithm** 4 employs a recursive function *DecryptNode()*, which was detailed in [1].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2705076, IEEE Access

Algorithm 3 Encryption (PK, T)

```
Let $\mathbb{A}$ be the access structure represented by $T$ rooted at node $R$;
Start from the root $R$ and choose a random $s \in \mathbb{Z}$ and set $q_R(0) = s$;
For each node $x$ in $T$ choose a polynomial degree $q_x$ and set the degree to $d_x = k_x - 1$;
for other nodes $x$ in $T$ do
    Set $q_x(0) = q_{parent(x)}(index(x))$
    Select $d_x$ randomly to define the polynomial $q_x$
end for
Let $Y$ be the set of leaf nodes in $T$, and the leaf nodes in $T$ describe the verification key $v_k$, and let $K = e(g_1, g_1)^{\alpha s}$;
The ciphertext is constructed as follows:
    $$\begin{aligned} CT = (T, C_1 = h_1^s, C_y = g_1^{q_y(0)} \\ C_y^1 = H(att(y))^{q_y(0)}, C_{v_k} = h_2^{q_{v_k}(0)} \\ C'_{v_k} = H(v_k)^{q_{v_k}(0)} : \forall y \in Y) \end{aligned} \quad (3)$$
Compute $\sigma = Sign_{s_k}(CT)$
The ciphertext is $C = (CT, \sigma)$
```

Algorithm 4 Decryption (SK, PK, C)

```
Verify the signature $\sigma$ using $v_k$;
Compute:
    $$F_{v_k} = \frac{e(C_{v_k}, H(v_k) \cdot g_1^{r/\beta_2})}{e(C'_{v_k}, h_2)} \quad (4)$$
for each node $x$ do
    if $x$ is a leaf node and $i \in S$ then
        $F_x = DecryptNode(CT, SK, x)$
            for all node $z$ that are children of $x$ do:
        $F_z = DecryptNode(CT, SK, z)$
    end if
end for
if $F_z \neq \perp$ then
    $F_x = \prod_{z \in S_x} F_z^{\triangle_{i, S'_x}(0)}$, where $i = index(z), S'_x = index(z) : z \in S_x$
end if
if The node is a root $R$ then
    $F_R = DecryptNode(CT, SK, R)$
    if $F_R == e(g_1, g_1)^{r \cdot qR(0)}$ then
        $F_R = \prod_{x \in \{R, v_k\}} F_x^{\triangle_{index(x), \{R, v_k\}}}$
    end if
end if
Compute $\frac{e(C_1, D)}{A}$ to get $K$
```

V. Analysis of the Proposed Protocol

In this section, we first show the feasibility and correctness of our protocol. Then we analyze the security properties of the proposed protocol by examining how it can resist several major attacks.

A. The Correctness of the Proposed Protocol

In this subsection, we illustrate that our protocol is correct and feasible. The fog node must first verify the signature σ on C using v_k to correctly decrypt the ciphertext. The verification is processed as follow:

$$\begin{aligned} F_{v_k} &= \frac{e(C_{v_k}, H(v_k) \cdot g_1^{r/\beta_2})}{e(C'_{v_k}, h_2)} \\ &= \frac{e(C_{v_k}, g_1^{r/\beta_2}) \cdot e(C_{v_k}, H(v_k))}{e(C'_{v_k}, h_2)} \\ &= \frac{e(h_2^{q_{vk}(0)}, g_1^{r/\beta_2}) \cdot e(e(h_2^{q_{vk}(0)}, H(v_k))}{e(H(v_k)^{q_{vk}(0)}, h_2)} \\ &= e(g_1^{\beta_2 \cdot q_{vk}(0)}, g_1^{r/\beta_2}) \\ &= e(g_1, g_1)^{r q_{vk}(0)} \end{aligned} \quad (5)$$

Then, a recursive function $DecryptNode(CT, SK, R) = e(g_1, g_1)^{r \cdot qR(0)} = e(g_1, g_1)^{rs}$ is executed on the root R of the subtree T. Let $A = e(g_1, g_1)^{rs}$, the decryption procedure to obtain the symmetric key is calculated as follow:

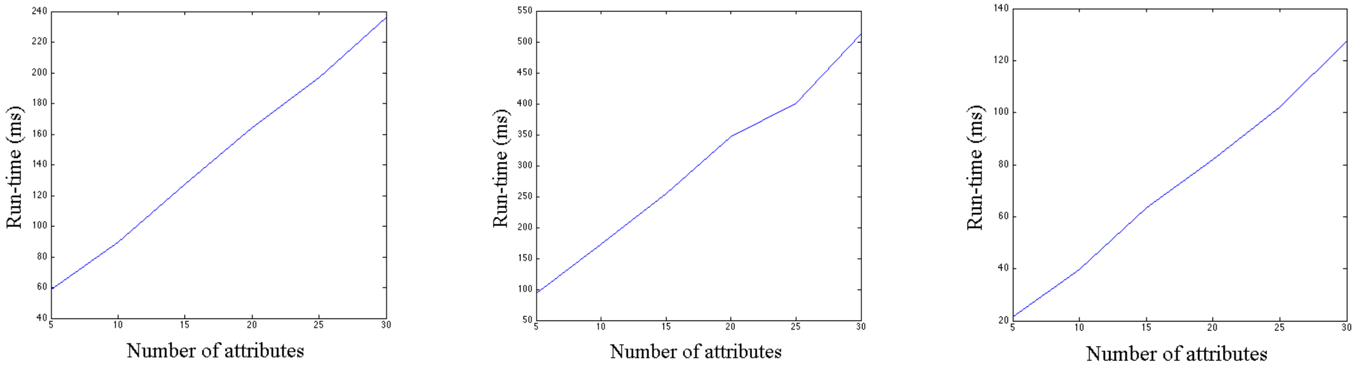
$$\begin{aligned} K' = \frac{e(C_1, D)}{A} &= \frac{e(h_1^s, g_1^{\alpha + r/\beta_1})}{e(g_1, g_1)^{rs}} \\ &= \frac{e(g_1, g_1)^{s(\alpha + r)}}{e(g_1, g_1)^{rs}} \\ &= e(g_1, g_1)^{\alpha s} = K \end{aligned} \quad (6)$$

If the verification succeeds in (5), which implies that the v_k has not been replaced and the output in (6) is correct. Otherwise, if the v_k has been replaced in (5), the result in (6) would be $\perp$.

B. Security Analysis

In this subsection, we analyze the security strength of our proposed protocol from the aspects of collusion attack resistance, message authentication, and unforgeability.

1) Collusion Attack Resistance: In the proposed scheme, we employ CP-ABE to guarantee the security of the shared key (session key). CP-ABE provides an access structure for each encrypted data, and requires only a subset of the attributes for decryption. Since the secret key involves a unique random number for each attribute in the access policy, CP-ABE can defend against collusion attacks. Thus illegal users can not obtain the exchanged shared key via collusion activities.



(a) The key generation time.

(b) The encryption time.

(c) The decryption time.

Fig. 4: The performance of our scheme.

2) Message Authentication: Assume that the cloud wants to send the symmetric key K to the fog nodes, which has the common attributes, the cloud encrypts K with **Algorithm** 3, then it broadcasts the encrypted message. When the fog nodes obtain the encrypted message, they need their private keys $SK = (D = g_1^{\alpha+r/\beta_1}, E = g_1^{r/\beta_2}, \forall j \in S : D_j, D'_j)$, which are computed by **Algorithm** 2. Meanwhile, the fog nodes obtain the cloud's verification key v_k. Then, the fog nodes verify the signature via **Algorithm** 4. If passed, the fog nodes decrypt the encrypted message to obtain the symmetric key K; otherwise, it is discarded.

3) Unforgeability: An adversary who wants to create a valid signature of a legal user must possess the user's private key. However, an adversary cannot infer the private key SK. On the other hand, it is impossible for the adversary to create a new, valid ciphertext and signature from another user's ciphertext and signature. If the adversary modifies the ciphertext of the shared key, the receiver can verify that the ciphertext is illegal using **Algorithm** 4. If the adversary colludes with other users to forge the ciphertext and signature, it cannot succeed because CP-ABE can defend collusion attacks. Thus we claim that our proposed scheme is unforgeable under chosen message attacks.

VI. Performance Analysis

In this section, we analyze the message size and communication overhead of the proposed scheme. Since the message size is directly related to the communication cost, we start from analyzing the message size.

A. Message size

We analyze the message size of the proposed scheme as follows. The Setup phase of our scheme involves the public key $PK = (\mathbb{G}_1, \mathbb{G}_2, g_1, g_2, p, H, h_1 = g_1^{\beta_1}, h_2 = g_1^{\beta_2}, e(g_1, g_1)^{\alpha})$ and master key $MK = (\beta_1, \beta_2, g^{\alpha})$, which result in a total size of $|\mathbb{G}_1| + |\mathbb{G}_2| + |g_1| + |g_1| + |p| + |H| + |h_1| + |h_2| + |e| + |\beta_1| + |\beta_2| + |g^{\alpha}| = 11|\mathbb{G}_1| + |H|$. In the Key Generation phase, the key generation server needs to generate the secret key $SK = (D = g^{\alpha+r/\beta_1}, E = g^{r/\beta_2}, \forall j \in S : D_j, D'_j)$ for the cloud. The total message size can be calculated as $|D| + |E| + ||S||(|D_j| + |D'_j|) = 2|\mathbb{G}_1| + 2||S|||\mathbb{G}_1| = (2+2||S||)|\mathbb{G}_1|$, where $||S||$ is the number of attributes. In the Encryption phase, the cloud encrypts the symmetric key to generate the ciphertext C and signature σ; thus the message size is $|CT| + |\sigma| = |C|$. In the Decryption phase, the fog node needs to store the symmetric key K, so the size is $|K|$.

TABLE I: The message size in Setup, Key Generation, Encryption, and Decryption phases at the sender and receiver.

	Setup	Key Generation	Encryption	Decryption
Message size	$11\|\mathbb{G}_1\| + \|H\|$	$(2 + 2\|\|S\|\|)\|\mathbb{G}_1\|$	$\|C\|$	$\|K\|$

Notes: $|\cdot|$ denotes the length of message; $||\cdot||$ denotes a number of the set.

B. Communication Overhead

The cloud and fog nodes exchange the shared key, which can be transmitted between them when needed; thus the communication overhead is mainly related to the size of the ciphertext.

TABLE II: Communication overhead of the fog node and the cloud.

	Communication cost
Key Generation	0
Encryption	$\|CT\| + \|\sigma\|$
Decryption	0

The Key Generation phase proposed in Section IV does not involve any message exchange, and thus its communication cost is zero. In the Encryption phase, the cloud sends the ciphertext CT and signature σ to the fog nodes, and thus the cost is $|CT| + |\sigma|$. The Decryption phase involves no communication, and thus the cost is zero. Table II summarizes the communication cost of our proposed scheme.

C. Comparison

In this subsection, we present a comparison study between our scheme and the traditional certificate-based scheme in terms of computational cost, transmission cost, and revocation issues.

To evaluate the impact of the computational overhead in our scheme and the certificate-based scheme, we are mainly concerned about the cryptographic operations: encryption and decryption. In a certificate-based scheme, the computational cost takes 7201.3 ms according to Hong and Lim in their work on biometric enabled X. 509 certificate. This cost is mainly due to the decryption operation that includes the verification phase for the certificate's signature. The certificate's signature cost grows linearly with the length of the certificate chain. In our scheme, the total computational cost is 638.9 ms. The major computational overhead occurred in the encryption phase due to the additional cost of the signature operation.

Another issue in the certificate-based scheme is the transmission cost. This scheme utilizes the certificate to bind the identity information to the keys. This means the certificate needs to be transferred alongside with the signature to verify it properly and thereby significantly increase the transmission overhead. Lastly, additional overhead results from checking the certificate's status and the certificate's validity period using either the Certificate Revocation List (CRL) or Online Certificate Status Protocol (OCSP). In fact, the most common revocation approach is the CRL which is required to download the CRL file to check the certificate's status. The size of a CRL file can vary between a few bytes to megabytes depending on the number of the revoked certificates and thus it adds a storage overhead.

In contrast, our scheme does not incur any transmission overhead because it does not need to exchange certificates or any identity information since the user's attributes are associated with the private key. Additionally, there is no need to download a file or communicate with a third party to check the certificate's status since each private key is correlated with an expiration date. In summary, our scheme is more efficient and feasible compared with the certificate-based scheme.

VII. Implementation

We run the experiment on Python under OS X 10 operating system with a 1.3 GHz Intel Core i5. We utilize the Charm cryptography library that wraps the Stanford Pairing-Based Cryptography (PBC), which is an open source library that performs the core mathematical functions of pairing-based cryptosystems. We test our algorithms under (SS512) elliptic curves with symmetric bilinear pairings and the numbers of attributes are chosen from 5 to 30.

Fig. 4 (a) shows the run time of **Algorithm** 2, where times are measured in milliseconds. The algorithm runs in linear time with the number of attributes that is associated with the number of issued keys. The performance of **Algorithm** 3 is illustrated in Fig. 4 (b) where the polynomial operations at the leaf nodes do not add a significant amount of time to the running time. The running time of **Algorithm** 4 is shown in Fig. 4 (c). It is slightly higher than the original algorithm due to the verification process. However, considering that our protocol provides signature and encryption with relatively trivial time in practice, our protocol is more desirable and feasible than the existing ones.

Briefly, the running times in both **Algorithm** 2 and **Algorithm** 3 are predictable as they depend on the number of attributes associated with the keys or the leaf nodes. In **Algorithm** 4, the performance depends on the number of available attributes and the access tree policy.

VIII. Conclusion

In this paper, we design an encrypted key exchange protocol to establish secure communications among a group of fog nodes and the cloud. In our protocol, we utilize the digital signature and CP-ABE methods to achieve the primary security goals: confidentiality, authentication, verifiability, and access control. We analyze the security of our protocol and show its correctness and feasibility. We also provide an implementation of our scheme. We further compare the proposed scheme with the certificate-based scheme and illustrate its efficiency.

In our future research, we will focus on the following directions. First, we intend to design a secure protocol with less computation overhead to make it suitable for IoT communications. Second, we will design an efficient access structure for fog computing and IoT devices.

Acknowledgment

Alrawais acknowledges the scholarship fund from the Ministryof Higher Education, Saudi Arabia and the College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Saudi Arabia; Alhothaily acknowledges the scholarship fund from Saudi Arabian Monetary Authority; Xiaoshuang Xing acknowledges the National Natural Science Foundation of China (61602062) and the Natural Science Foundation of Jiangsu Province (BK20160410).

References

[1] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE symposium on security and privacy (SP'07)*. IEEE, 2007, pp. 321–334.

[2] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 195–203.

[3] A. Lewko and B. Waters, "Unbounded hibe and attribute-based encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2011, pp. 547–567.

[4] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2010, pp. 62–91.

[5] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in *Annual Cryptology Conference*. Springer, 2010, pp. 191–208.

[6] L. Cheung and C. Newport, "Provably secure ciphertext policy abe," in *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 456–465.

[7] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 2010, pp. 735–737.

[8] Z. Wan, J. Liu, and R. H. Deng, "Hasbe: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE transactions on information forensics and security*, vol. 7, no. 2, pp. 743–754, 2012.

[9] D. Huang, Z. Zhou, L. Xu, T. Xing, and Y. Zhong, "Secure data processing framework for mobile cloud computing," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*. IEEE, 2011, pp. 614–618.

[10] J.-M. Do, Y.-J. Song, and N. Park, "Attribute based proxy re-encryption for data confidentiality in cloud computing environments," in *Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on*. IEEE, 2011, pp. 248–251.

[11] L. Xu, X. Wu, and X. Zhang, "Cl-pre: a certificateless proxy re-encryption scheme for secure data sharing with public cloud," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*. ACM, 2012, pp. 87–88.

[12] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE transactions on parallel and distributed systems*, vol. 24, no. 1, pp. 131–143, 2013.

[13] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Infocom, 2010 proceedings IEEE*. Ieee, 2010, pp. 1–9.

[14] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE transactions on knowledge and data engineering*, vol. 25, no. 10, pp. 2271–2282, 2013.

[15] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the internet of things: Security and privacy issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017.

[16] 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 *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*. ACM, 2013, pp. 15–20.

[17] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of internet of things," *IEEE Transactions on Cloud Computing*, 2015.

[18] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular adhoc network with fog computing," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. IEEE, 2015, pp. 1202–1207.

[19] M. Al Faruque and K. Vatanparvar, "Energy management-as-a-service over fog computing platform," *IEEE Internet of Things Journal*, 2012.

[20] Y. Cao, S. Chen, P. Hou, and D. Brown, "Fast: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation," in *Networking, Architecture and Storage (NAS), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2–11.

[21] A. Beimel, *Secure schemes for secret sharing and key distribution*. Technion-Israel Institute of technology, Faculty of computer science, 1996.

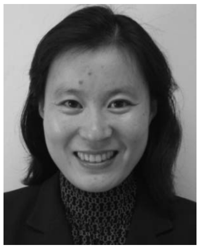
[22] M. C. Gorantla, C. Boyd, and J. M. G. Nieto, "Attribute-based authenticated key exchange," in *Australasian Conference on Information Security and Privacy*. Springer, 2010, pp. 300–317.

Arwa Alrawais received her M.S. degree in Computer Science from The George Washington University in 2011. Currently she is a Ph.D candidate at the Department of Computer Science, The George Washington University, Washington DC. Her current research interests include network security, wireless and mobile security, and algorithm design and analysis. She is a student member of IEEE.

Abdulrahman Alhothaily is a Ph.D. candidate in Computer Science in the field of Computer Security and information assurance at The George Washington University. He holds a Masters degree in Computer Science from The George Washington University, and a Bachelors degree in Computer Engineering from King Saud University. His current research interests include Payment Security, Fraud, Wireless and Mobile Security, and Security Engineering. Previously, Abdulrahman was involved in designing and implementing the security of critical payment systems at the Saudi Arabian Monetary Authority as a Security Engineer. He won a number of awards, including the "Best Student Award" from the George Mason University; and a "Ph.D Scholarship" awarded by the Saudi Arabian Monetary Authority.

Chunqiang Hu received his M.S degree and first PhD degree in computer Science and Technology from Chongqing University, Chongqing, China, in 2009 and 2013, respectively. He obtained his B.S. degree in Computer Science and Technology from Southwest University, Chongqing, China, in 2006. He was a visiting scholar at The George Washington University from January 2011 to December 2011, and got his second PhD degree in Computer Science, The George Washington University, Washington DC in 2016. Currently he is a faculty member in School of Software Engineering, Chongqing University, honored with the Hundred-Talent Program. His research interests include privacy-aware computing, big data security and privacy, wireless and mobile security, applied cryptography, and algorithm design and analysis. He is a member of IEEE and ACM.

Xiaoshuang Xing received her BA degree from the North China Electric Power University, Baoding, China. She graduated from Beijing Jiaotong University with a PhD degree in communication and information systems in 2014. Her primary research interests include spectrum prediction in cognitive radio networks, physical layer security, resource management in cognitive radio networks, and social ad hoc networks.



Xiuzhen Cheng received her MS and PhD degrees in computer science from the University of Minnesota – Twin Cities, in 2000 and 2002, respectively. She is a professor in the Department of Computer Science, The George Washington University, Washington DC. Her current research interests include privacy-aware computing, wireless and mobile security, and algorithm design and analysis. She worked as a program director for the US National Science Foundation (NSF) for six months in 2006 and joined the NSF again as a part-time program director in April 2008. She received the NSF CAREER Award in 2004. She is a fellow of IEEE.