IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 26, 2020, accepted April 29, 2020, date of publication May 6, 2020, date of current version May 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992460

Access Control in Fog Computing: Challenges and Research Agenda

MOHAMMED A. ALEISA[1,2], (Member, IEEE), ABDULLAH ABUHUSSEIN[3], (Member, IEEE), AND FREDERICK T. SHELDON[1], (Senior Member, IEEE)

[1]Department of Computer Science, University of Idaho, Moscow, ID 83844, USA
[2]Department of Computer Science, Majmaah University, Al-Majmaah 11952, Saudi Arabia
[3]Department of Information Systems, St. Cloud State University, St. Cloud, MN 56301, USA

Corresponding author: Mohammed A. Aleisa (alei3598@vandals.uidaho.edu)

ABSTRACT Fog computing is an intermediate computing layer that has emerged to address the latency issues of cloud-based Internet of things (IoT) environments. As a result, new forms of security and privacy threats are emerging. These threats are mainly due to the huge number of sensors, as well as the enormous amount of data generated in IoT environments that needs to be processed in real time. These sensors send data to the cloud through the fog computing layer, creating an additional layer of vulnerabilities. In addition, the cloud by nature is vulnerable because cloud services can be located in different geographical locations and provided by multiple service providers. Moreover, cloud services can be hybrid and public, which exposes them to risks due to their infinite number of anonymous users. Access control (AC) is one of the essential prevention measures to protect data and services in computing environments. Many AC models have been implemented by researchers from academia and industry to address the problems associated with data breaches in pervasive computing environments. However, the question of which AC model(s) should be used to prevent unauthorized access to data remains. The selection of AC models for cloud-based IoT environments is highly dependent on the application requirements and how the AC models can impact the computation overhead. In this paper, we survey the features and challenges of AC models in the fog computing environment. We also discuss the diversity of different AC models. This survey provides the reader with state-of-the-art practices in the field of fog computing AC and helps to identify the existing gaps within the field.

INDEX TERMS Access control, attribute-based encryption, authorization, data modification, data ownership, fog computing, IoT devices, security, privacy, malicious activities, outsourcing decryption, outsourcing encryption.

I. INTRODUCTION

Cloud computing provides IoT (Internet of things) environments with a facility for computation and storage. However, cloud computing requires a high latency due to its distance from the end user [1]. Additionally, the data generated from IoT devices takes time to be computed in the cloud. As the number of IoT devices increases, the amount of data generated will also increase. This huge amount of data aggregated from devices located far away from the cloud must be transferred with low latency. To solve this issue, fog computing emerged.

Fog computing serves as a middle layer between cloud and IoT devices to solve the problem of high data transfer latency. To meet the high processing demand, the huge number of sensors in IoT environments send data through fog nodes rather than directly to the cloud. Smart cities and smart grids are examples of systems in which fog computing can be found between the smart devices and the cloud [2], [3]. This additional layer (i.e., fog computing) can introduce new vulnerabilities since it expands the attack surface on which threats such as data loss and breaches can occur [4]. In addition, several threats, such as malicious fog nodes [5], man in the middle attacks [6], malicious insider threats [4], and denial of service attacks [4], arose in fog computing environments. For instance, in fog computing environments, attackers may seek infinite processing or storage in fog devices, which prevents users from accessing fog device(s) [7].

Access control (AC) is one of the crucial defense frontlines to maintain users' security and privacy, as well as to

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

protect data and services from unauthorized access. Due to the increase in the number and type of threats, it is essential to have effective AC models in fog computing environments. Cloud computing and fog computing are being used in many domains to provide support to IoT environments. This is known as cloud-fog-IoT architecture. To ensure the appropriateness of the AC strategies in the cloud-fog-IoT architecture, it is important to identify the requirements of the application for which this architecture is used. Application requirements in fog computing include, but are not limited to, scalability, mobility, and heterogeneity [8]. Thus, it is important to select AC models that meet these fog application requirements. Moreover, choosing one of the AC strategies over the others can have a negative impact. For example, it may significantly increase the computation overhead in fog nodes due to the multiple operations involved in AC models, such as file encryption, ciphertext decryption, and distribution of attributes [9]–[11]. On the other hand, using more than one AC model in fog nodes can cause additional heavy computation on fog nodes due to the heavy operations used in controlling access. Therefore, outsourcing part of the operations when implementing AC models for fog nodes becomes crucial. The aforementioned reasons demonstrate the need for dynamic and more efficient AC models. It is also important to appropriately select AC models to protect the cloud-fog-IoT architecture. In this paper, we survey AC models in fog computing, present their challenges, and identify gaps for future research.

The remainder of this paper is structured as follows: In sections 2 and 3, we discuss fog computing and AC comprehensively. In section 4, we present the state of the art in the field of AC in fog computing. In section 5, we discuss some security and privacy issues related to AC in fog computing. In section 6, gaps in the field are identified and discussed. Finally, we conclude this work in section 7.

II. WHAT IS FOG COMPUTING?

Fog computing is defined as an intermediate layer between the cloud and IoT devices [12]. Figure 1 presents a classic fog computing-aided IoT environment. Fog computing extends the cloud services to the edge of the network, near IoT devices, to reduce the latency and network congestion. Low latency is a desired quality in today's applications, such as emergency responses in the medical domain, and fog computing guarantees low latency by providing real-time processing capabilities for the transferred data [12]. According to Cisco [13], fog computing is the place where IoT data is analyzed near the IoT devices that generate and process data. A typical fog computing environment consists of nodes connected to IoT devices. These nodes are referred to as fog nodes.

Fog nodes can be deployed anywhere within the network connection. Fog devices can be any device that has computing, storage, and network connectivity. According to NIST [14], fog computing is an intermediate layer that allows global access to several IoT devices. The environment of fog computing enables the deployment of distributed applications and services [15], [16].

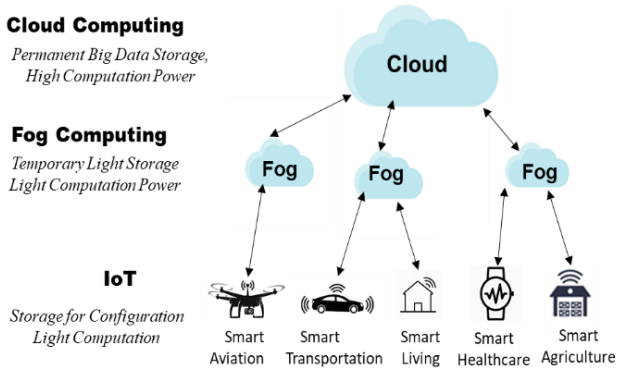

FIGURE 1. Fog computing environment.

A. WHY FOG COMPUTING?

The fog computing layer between the cloud and IoT devices has valuable functionalities:

1) MOVE CLOUD CONTENT CLOSER TO IoT

By bringing cloud content closer to IoT devices, fog computing solves the delay issues in time-sensitive applications in which decisions must be made in a timely manner.

2) SAVE NETWORK BANDWIDTH

Since not all data should be transferred to the cloud for processing, using a fog layer between the cloud and the IoT devices helps to save network bandwidth. In this case, fog computing can better handle managing and controlling data processing, transfer, privacy, and security. This will also reduce operating expense.

3) BRING STORAGE CLOSER TO IoT

This functionality of fog computing is essential because it places temporary storage closer to IoT devices which have limited storage capability. Fog computing serves as a temporary storage location for the data aggregated from IoT devices, whereas the cloud stores the data permanently.

4) BRING COMPUTATION POWER CLOSER TO IoT

In the cases in which data gathered from IoT devices require immediate processing, fog computing can serve as a processing facility that is located closer than the cloud. Fog computing, in this case, will take care of quick and small workloads. Big data analytics will still be handled by the cloud.

5) PROTECT IoT DATA

Although fog computing expands the cloud-fog-IoT architecture attack surface, this additional computing layer, with its storage and computing capability, can be utilized to host and run automated monitors to detect threats to IoT. It can also

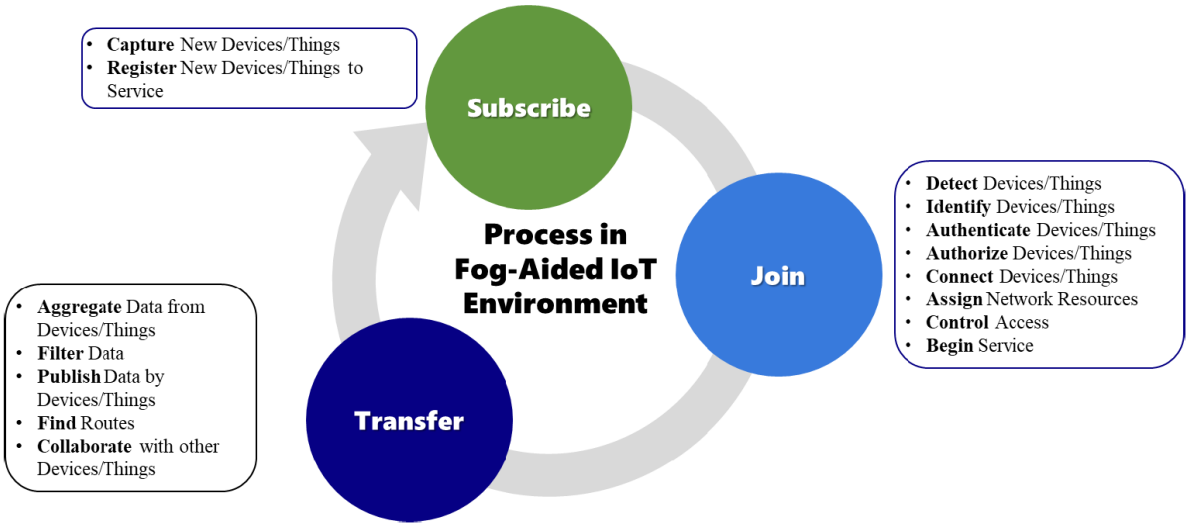

FIGURE 2. The subscribe, join, and transfer phases in a fog-aided IoT environment with events occurring in each phase.

be used to fine grain AC to avoid over- and underexposure of authorization.

B. FOG COMPUTING LAYERS

Several fog computing architectures have been proposed in [8], [12], [17]–[19]. Commercial architectures of fog computing have also been developed for commercial fog devices [18]. The architecture of a standard fog computing environment consists of several layers:

1) PHYSICAL LAYER

This layer represents all fog hardware devices that send and receive data to and from IoT devices. These devices can be virtual or physical devices, such as virtual and physical network routers.

2) MONITORING LAYER

This layer is responsible for detecting and logging performance and security-related flaws in IoT devices and/or fog nodes. For example, this layer can select a fog node based on criteria such as throughput, congestion, etc., and detect malicious activities against fog nodes or IoT devices.

3) PROCESSING LAYER

This layer is responsible for analyzing and filtering the data collected from IoT devices. As the number of IoT devices increases, the amount of data also increases. Therefore, processing this enormous amount of data can be challenging. Fog nodes usually have light-to-medium-weight processing capability. Intensive processing is usually performed in the cloud.

4) STORAGE LAYER

This layer is responsible for storing data generated from IoT devices. IoT devices have limited storage capability, so fog computing provides a temporary storage service for IoT data. Long-term storage and storing historical data are usually handled by the cloud.

5) SECURITY LAYER

This layer maintains the security objectives (i.e., confidentiality, integrity, and availability) in the fog nodes. The security layer is where all controls and measures are applied to detect and prevent threats, as well as to respond to security incidents. For example, encryption and decryption of data received and sent by fog devices is a security measure handled by the fog computing layer to maintain confidentiality and is considered a prevention technique. In another example, fog nodes may be used to balance the load directed to IoT devices based on throughput or congestion in cases of denial of service (DoS) attacks. The objective of this security measure is to maintain availability by responding to a DoS incident.

6) APPLICATION LAYER

This layer includes the applications and protocols responsible for networking (such as routing) and load balancing (such as routing tables and Hypertext Transfer Protocol (HTTP) and MQ Telemetry Transport (MQTT) protocols) [20].

C. FOG-AIDED IoT PROCESS PHASES

In fog-aided IoT environments [21], [22], a subscribed IoT node may request to join a fog network before it can collect and publish data. This model is known as the **publishing/subscribing model**. In another model, a fog-aided IoT network finds an IoT node and requests to add it to the network in order to collect and publish data. This is known as the **request/response model**. When the IoT node joins, it is assigned network resources and can then start communicating and operating as a component of the IoT environment. Figure 2 shows a three-phase process in a

fog-aided IoT environment. The process assumes a fresh start of an unsubscribed node. Thus, initially, a node needs to be *subscribed* on-demand before it can *join* the fog network and *transfer* data within the network (by aggregating and publishing data, for example).

1) SUBSCRIBE PHASE

When a new IoT device wants to connect to a fog device, the fog device captures the new IoT device that needs services and registers it to the requested services.

2) JOIN PHASE

Fog devices detect new IoT devices that request services, and each IoT device is asked to show its identity before joining the fog network. To avoid security issues from malicious IoT devices, the new IoT devices should be authenticated first. After that, the authenticated devices may request access to fog devices to obtain authorization for the services provided. This is where AC models are applied. When the authentication and authorization operations are complete, different groups of IoT devices become connected to the corresponding fog devices with access to network resources and service may begin.

3) TRANSFER PHASE

In this phase, fog devices start aggregating data from IoT devices and/or send tasks to them. When the amount of data collected is huge, fog devices filter data received from IoT devices before processing it or sending it to the cloud. Load balancing strategies can be used to send workload to free fog nodes when a fog node is overwhelmed with tasks.

Fog computing receives data from IoT devices and then processes it or sends it to cloud storage. Fog computing can interact with all three types of cloud services (Software as a Service [SaaS], Platform as a Service [PaaS], and Infrastructure as a Service [IaaS]) [14]. To the best of our knowledge, there is no standard architecture for fog computing. Several commercial platforms, including ParaDrop and Cloudlet, have been proposed [23]. ParaDrop is a fog computing platform based on wireless routers using operating system-level virtualization. A cloudlet is a mobility enhanced small-scale cloud infrastructure that is located at the edge of the Internet and can act as a fog layer.

There are several areas where fog computing can be utilized, such as smart cities, smart vehicles, smart grids, and mobile healthcare [24]. Figure 3 shows a taxonomy of fog computing applications. In the healthcare domain, data are generated by thousands of sensors that require low latency and real-time processing demand. Fog nodes can be a feature to support the scalability of patient monitoring.

D. CHARACTERISTICS OF FOG COMPUTING

As the number of IoT devices increases, handling the data generated by IoT devices and transferring it to the cloud may turn out to be challenging. Therefore, fog computing emerged to address these challenges by processing the data at the edge of the network (close to the IoT devices), which results in reduced latency. Table 1 depicts the differences between fog computing and cloud computing in terms of the following common characteristics:

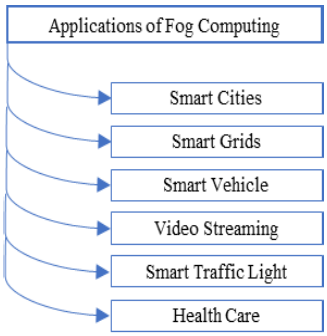

FIGURE 3. Fog computing applications.

TABLE 1. Characteristics that differ between fog computing and cloud computing.

Characteristics	Fog Computing	Cloud Computing
Architecture	Decentralized	Centralized
Latency	Low	High
Location Awareness	Yes	No
Mobility	Supported	Limited
Geographic location	Yes	No
Delay	Low	High
Scalability	High	Limited
Deployment	At the edge of the network	Network Core

1) LATENCY

Some transfer delay between IoT devices and the cloud can be tolerated, depending on the requirements and the nature of the application. However, for medical applications or in case of emergency events, the data are very time-sensitive. The latency will be high in the cloud because the distance between IoT devices and the cloud is long. Thus, computing the data in the cloud will cause a high latency. However, fog computing reduces that latency by bringing data to the edge of the network and closer to end users to meet the high processing demand [25].

2) SCALABILITY

As the number of IoT devices increases, it is difficult for the cloud to handle the heavy computation and bandwidth overhead of these devices. Fog computing can solve this issue by distributing serval fog nodes that can reduce the heavy computation and support hierarchal scalability when the number of IoT devices increases [25].

3) LOCATION-AWARENESS

Since the cloud is far from IoT devices, sending location information may push heavy workloads toward the cloud when the number of IoT devices is high. Therefore, having fog nodes closer to the IoT devices to manage and control

traffic sent to the cloud and to support geographic location becomes necessary [25].

4) MOBILITY

Fog computing supports the feature of mobility. Per Cisco, any device that has computing, storage, and network connectivity can be a fog node [13]. In fog computing, a fog node can be any mobile device, such as smart vehicles or smart phones, or any static device, such as traffic cameras in smart city devices [26].

5) GEOGRAPHIC LOCATION

A fog computing layer may consist of a number of distributed fog nodes that are deployed in different locations [8]. As previously mentioned, fog computing supports the feature of geographic location, and distributed fog nodes can track the locations of IoT devices to support their mobility. The applications and services of fog computing are decentralized and can process and store data from end devices. Therefore, the massive amounts of data generated by IoT devices will be processed faster in decentralized fog computing than in centralized cloud computing.

6) HETEROGENEITY

The fog computing layer consists of two components: the physical node and the virtual node. Physical nodes include physical sensors and routers, while virtual nodes include virtual sensors and virtual load balancers. These physical or virtual nodes may have different operating systems and may be used to run different applications. Therefore, heterogeneity in fog nodes is desirable to make these devices interoperable [8], [26].

7) BANDWIDTH

Fog computing can process the data created by IoT devices at the edge of the network, close to the end user, rather than sending it to the cloud. Therefore, fog computing efficiently saves the bandwidth by computing and storing the data locally. As the number of IoT devices increases, more data may be generated and collected. Therefore, an architecture of distributed fog nodes addresses this problem by computing the enormous amount of data locally instead of transmitting it to the cloud. This, in turn, reduces network traffic and saves bandwidth [26].

III. ACCESS CONTROL OVERVIEW

AC is based on a data access policy (e.g., HIPAA [27]) that determines what privileges are granted to which roles within the various operational scenarios. In other words, the user should first be authenticated to access the system. Then, the user can request access to the system resources and be authorized by the system administrator [28]. There are multiple AC models, such as Discretionary Access Control (DAC), Mandatory Access Control (MAC), Role-Based Access Control (RBAC), and Attribute-Based Access Control (ABAC) [28]. Figure 5 shows a taxonomy of AC models in

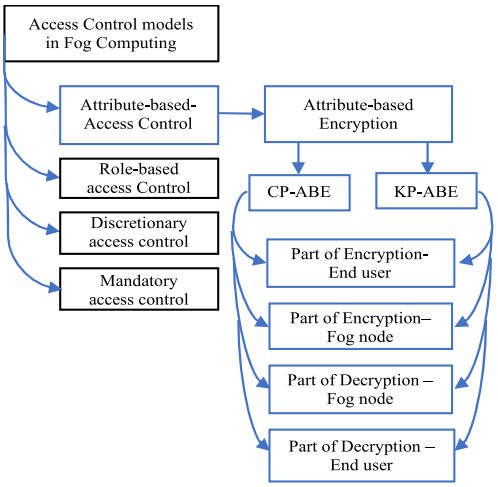

FIGURE 4. Access control models in fog computing.

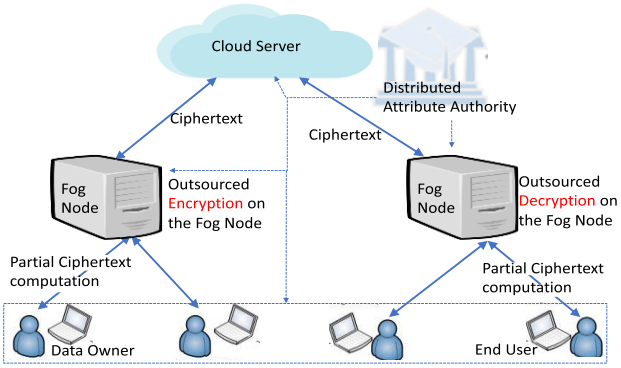

FIGURE 5. Layers of fog computing.

fog computing. Attribute-Based access control (ABAC) could be an appropriate model to deal with fog computing, as the owner of the data in the fog layer can define the AC policies for users to achieve the authorization [18]. Here, we summarize AC models:

A. ACCESS CONTROL MODELS

1) ATTRIBUTE-BASED ACCESS CONTROL (ABAC)

One of the most well-known AC models is attribute-based access control (ABAC) [28]. This model has three key elements: attributes, a policy model, and an architecture model. Attributes are features that define a user, a resource, or an environmental condition. In fog computing environments, there are four components that interact with fog devices: IoT devices, the data owner, users, and attribute authority. When fog nodes receive data from IoT devices for high processing demand, users may want to access the data. Therefore, a set of attributes will be provided to the users and resources to grant the authorization for access. Attributes can be a username, a job, a resources owner, and/or an environment's time or date created or last accessed. AC policies are defined by the owner of the data, which could be an organization or an individual. AC policies are rules specified by the owner of the data within

the organization. These rules can be defined based on users' behavior [28].

2) DISCRETIONARY ACCESS CONTROL MODEL (DAC)

This AC model controls access based on the identity of the users who request the access. Any authorized entity can grant access rights, such as read, write, and/or view, to others. This model is less secure and known to cause management overhead in the environment of fog computing [28].

3) ROLE-BASED ACCESS CONTROL MODEL (RBAC)

In this model, AC is defined based on the role of the user in the organization, such as students, faculty, and staff in universities and colleges. Therefore, access rights are assigned to the roles instead of the users. Some users may have more than one role within the organization. In this situation, AC policies for each role are applied and may overlap. In this model, AC rights that the owner of data would grant to users are view, read, update, and/or write. When a user requests to access data, the user's role is compared to the access policy that is predefined by the owner of the data and access is granted accordingly [28].

4) ACCESS POLICY ACCESS CONTROL MODEL (APAC)

A policy is a set of rules that is pre-defined by the data owner. The owner of the data can be an organization or an individual that sets up the policy for access to their resources. These rules may consist of authorized behaviors that are defined by the owner of the data and meet the data owner's security objectives. Each user has one or more identifying attributes. Access policies consist of attributes that define every user's accessibility to the resources. These attributes are written in access policy in multiple levels and may be connected by a logical expression such as AND or OR [28].

5) IDENTITY-BASED ACCESS CONTROL (IBAC)

There are three elements that interact in AC: subject, object, and access rights. The subject is an active entity and can be a user or an application requesting access to a resource(s). The object is a passive entity and can be a resource for which access needs to be controlled. The access rights are the method by which a subject may access an object. The access rights consist of several operations, such as read, write, delete, and search. This model manages any access by a subject to an object through access rights. This model is based on the identity of the subject and an object identifier [29].

6) TASK-BASED ACCESS CONTROL (TBAC)

In this model, a task is considered a subrole for a subject. When the task of a subject satisfies the roles involved in the task, the subject is granted access to an object [29].

7) RULE-BASED ACCESS CONTROL (RBAC)

In this model, rules are defined such that a subject can access an object through satisfying these rules. As in DAC, access control lists (ACL) are associated with each object and include access rights of a subject to gain access to an object. When a user tries to access a resource, the system checks the rules in the ACL for that resource. Then, if rules are satisfied, the user gains access to a resource. For example, students may access a course website only at a certain time of day [30].

8) MANDATORY ACCESS CONTROL (MAC)

This model manages access based on comparing security labels with security clearances. The security labels are allocated to each object, such as a resource, and indicate how important the system resources are. The security labels consist of two components: (1) a classification component (e.g., top secret) and (2) a category component which declares the level to which the object is available. The subject, which is a user, has a classification and a category. When a user tries to access a resource using MAC, the system checks the classifications and category of the user and compares it to the security labels of the resource. Then, if the classification and the category of the user match the security labels of a resource, access is permitted. Otherwise, access is denied [30].

B. ACCESS CONTROL REQUIREMENTS IN FOG COMPUTING ENVIRONMENTS

Although we thoroughly surveyed the AC models, some of the AC models mentioned earlier are not used for fog computing. Thus, we identified the requirements for adopting and applying AC in fog computing. The requirements necessary to maintain efficiency in fog computing are as follows:

1) AC models use operations such as building access policy, which may cause computational overhead on the side of the IoT device. Since IoT devices have limited resources, the computational overhead can be taken care of by the closest fog node [18].

2) AC models should support the creation, deletion, and revocation of an AC policy. For example, what techniques should be used to update the system when policy is revoked [18]? In fog-based environments, the emerging fog layer further exposes the user data and applications since it is an additional attack point. This necessitates the application of an AC model that enables policy creation, deletion, and revocation at the cloud, fog, and IoT device levels.

3) Since the IoT devices are resource-limited, it is essential to restrict some resources from being accessed when the number of IoT devices increases [18].

4) AC models should support the revocation of attributes. This is important (1) to prevent the user whose attribute is revoked from being able to decrypt the new encrypted data (i.e., backward security) and (2) to enable the newly subscribed user whose attribute is satisfied and valid to decrypt the newly published encrypted data (i.e., forward security) [31].

5) Since the fog layer is supposed to be close to the IoT devices to solve the latency issue, the time required to decide whether the access policy is satisfied should be low and reasonable. If the user's attribute satisfies the access policy,

the policy decision's response time should be low. In addition, execution cost, networking cost, and deployment cost of the fog-based AC models should also be reduced since the fog layer is close to the IoT devices [18], [32].

6) As the number of IoT devices increase, the fog nodes will also increase. Therefore, multiple attribute authorities are needed to support scalability of fog nodes and IoT devices. Thus, AC models should support multiple attribute authorities [33]

7) Selecting an AC model in fog computing depends on the application requirements, which can impact the computation overhead. Some of the AC models mentioned earlier can be encryption-based, which results in additional operations (e.g., encryption and decryption). Thus, it is important to decide whether data encryption is required and mandated as an application requirement before selecting the AC model [8].

8) As the number of IoT devices increases, the data generated by these IoT devices will also increase. It is important to utilize an AC model that gives data owners more flexibility to underexpose/overexpose their data. Therefore, fine-grained AC becomes an essential requirement [34].

IV. STATE OF THE ART IN FOG ACCESS CONTROL

Attribute-based encryption (ABE) has been extensively studied [35], [36] and used in several schemes. ABE is a type of public encryption that is dependent on the attributes of the users and resourced accessed (i.e., APAC). Users who want to obtain access obtain secret keys that reflect their attributes, and the ciphertext has attributes that are encrypted according to policies created by data owner. Then, if the user's attributes satisfy the attributes embedded in the ciphertext, the user can decrypt the ciphertext and obtain access to the plaintext. In healthcare applications, a patient can define access policies by using logical expressions such as AND or OR and encrypt their personal health record according to the defined policies. The doctor can decrypt the personal health record if the doctor's attributes satisfy the access policies embedded in the ciphertext. There are two types of attribute-based encryption (ABE): Ciphertext-Policy Attribute-Based Encryption (CP-ABE) and Key Policy Attribute-Based Encryption (KP-ABE), as shown in Figure 4.

The existing schemes CP-ABE and KP-ABE have a number of operations to handle encryption and decryption, which cause a heavy computation overhead due to the resource constraints at the end users' side. Figure 4 shows a system model of applying ABE in fog computing. To decrease the heavy computation at the end users' side, serval works that outsource the encryption and decryption operations to the near fog nodes have been proposed, as shown in Figure 5 and Table 2.

The authors of [10] proposed an AC CP-ABE scheme that outsources the heavy computation of encryption and decryption to fog nodes. This makes the number of attributes in access policy and secret keys independent from the encryption/decryption computation. This scheme uses the intermediate fog layer to reduce the computational overhead for the data owner or end users. As shown in Figure 5, this scheme has five types of entities: (1) a cloud service provider (CSP), (2) fog nodes, (3) the data owner, (4) the end user, and (5) the key authority. The authors assumed that CSP and fog nodes are trusted in the scheme. The data owner has files that need to be encrypted before being sent to the cloud. Each data owner is responsible for defining access policy and generating part of the ciphertext of the encrypted file before sending it to a fog node. A fog node is responsible for the creation of the other part of the ciphertext of the encrypted file. Then, the whole ciphertext is uploaded to the CSP. The key authority oversees user registration and the creation of secret keys for users. The secret keys reflect each user's attributes. The end user is the user on the other side who wants to gain access to the encrypted data stored in the cloud. If the end user's attribute set satisfies the access policy embedded in the ciphertext, the ciphertext will be uploaded from the cloud by the fog node. The fog node will then decrypt part of the ciphertext. The other part of ciphertext is decrypted by the end user. Updated users are those whose attributes are updated by the key authority, and non-updated users are those whose attributes are not yet updated.

The authors in [37] proposed a framework that secures the sharing of personal health records (PHR) in a cloud computing environment. Their framework supports the scalability feature in cloud computing. The patients can encrypt their PHR files so that only authorized users can decrypt and access them. The presented framework classifies users according to security domains as (1) public domain and (2) private domain. The public domain includes the professional users who are managed distributively by multiple attribute authorities. The attribute authority can control several attributes for all users in the public domain, and each user should be able to reach more than one attribute authority to get his/her attributes. Multiple Authorities Attribute-Based Encryption (MA-ABE) is used by public domain users, such as physicians, so that the user (i.e., physician) attributes represent the professional role of the user in the healthcare domain. In the public domain, patients (i.e., owners) defines access policies for their PHR files based on the professional roles of the users in that domain. On the other hand, the private domain includes those close to the PHR owner, such as family members, and access rights are assigned by the PHR owner to all users in the private domain. KP-ABE is used with private domain users such as close friends or family members so that a patient can control secret keys and access rights for their PHR files. PHR files are encrypted using ABE, so a PHR owner can easily permit users from two domains to access the files. This framework tackled the key and attribute management issues for all users by dividing them into two types of security domains and supported to the scalability features.

The authors in [38] proposed a framework, Privacy Preserving Cipher Policy Attribute-Based Encryption (PP-CP-ABE), to secure data inquiry in mobile cloud computing. This framework protects the collected data from mobile devices. Therefore, outsourcing the heavy

TABLE 2. Comparison of features in different AC schemes.

	Features						
Schemes Authors	Search based Keywords	Fog Computing	Multiple Authorities	Outsourcing Encryption	Outsourcing Decryption	Access Control	Citation
Zhang [10]		●		●	●	●	59
Zhou [38]				●	●	●	226
Asim [39]				●	●	●	16
Mao [40]					●	●	63
Zuo [41]		●			●	●	64
Alrawais [42]		●				●	60
Li [37]			●			●	1033
Yang [31]			●			●	2
Wang [43]				●	●	●	29
Li [44]	●				●	●	120
Huang [34]		●		●	●	●	48
Sun [11]	●	●	●	●	●	●	3
Miao [9]	●	●		●	●	●	26
Vohra [33]		●	●		●	●	2
Fan [45]		●	●	●	●	●	32
Xu [46]		●	●	●	●	●	9
Xue [47]		●	●	●	●	●	21

computation of encryption and decryption operations to CSPs can be achieved using PP-CP-ABE. Attribute-Based Data Storage (ABDS) is also proposed to decrease the data management overhead caused by CSPs. In the system architecture, there are three service providers: (1) encryption service provider (ESP), (2) decryption service provider (DSP), and (3) storage service provider (SSP). The ESP presents the encryption service to data owners without disclosing the data, the DSP presents the decryption service to users without exposing the data, and the SSP stores the encrypted data. Mobile devices can outsource the heavy operations of encryption and decryption to ESP and DSP using PP-CP-ABE.

The authors in [39] proposed an ABE scheme that supports the outsourcing encryption of a host and the outsourcing decryption of a user. The host and users are using mobile devices that have limited computational power, and the ABE scheme requires several operations of encryption and decryption. Therefore, the proposed ABE scheme uses two semi-trusted proxies. The first semi-trusted proxy is used to outsource the computation of encryption operations, and the second semi-trusted proxy is used to outsource the heavy computation of decryption operations. Thus, the computational overhead of mobile devices can be reduced. In the encryption stage, the data owner encrypts part of the message and the proxy encrypts the remaining part. This occurs according to the access policy that is defined by a host with set of attributes. In the decryption stage, the proxy compares the predefined access policy by the host with the user's attributes. If the user's attributes satisfy the access policy embedded in the encrypted message, the proxy decrypts part of the ciphertext and transforms the ciphertext to ElGamal ciphertext style. The ElGamal ciphertext can then be decrypted by the user. This scheme can help to relieve the computation overhead in constrained devices.

The authors in [31] proposed a Data Access Control for Multi-Authority Cloud Storage (DAC-MACS) system. A multi-authority CP-ABE scheme with support for an attribute revocation method that achieves both forward and backward security was also proposed. The system model has several entities: (1) a global trusted Certificate Authority (CA), (2) multiple Attribute Authorities (AAs), (3) a cloud server, (4) data owners, and (5) users. The CA is responsible for registering all users and AAs in the system. Each user is assigned a global unique identity by the CA. Every AA is responsible for managing and distributing secret keys that reflect users' attributes or roles. The cloud server stores the owner's data after it has been encrypted and allows users to access the data if the user's attributes satisfy access policies defined in the ciphertext. Data owners can define access policies and encrypt their data according to predefined access policies. Each user is allowed to decrypt the data stored in the cloud server if their secret keys issued by multiple AAs satisfy access policy embedded in the ciphertext.

The authors in [34] proposed a data AC scheme with the ciphertext update based on CP-ABE and an Attribute-Based Signature (ABS) in fog computing. This scheme has

delegated most operations of encryption to the data owner, decryption to IoT devices, and signing to update the ciphertext to fog nodes. Therefore, fog nodes perform heavy computation.Since IoT devices are resource-constrained, they can outsource their heavy computation to fog nodes. This scheme consists of five entities: attribute authority, CSP, fog nodes, data owner, and end user, as shown in Figure 2. The data are first encrypted with a symmetric encryption algorithm by the data owner, who also then defines an access policy and update policy. The access policy is used for end users to decrypt the data when their attributes are satisfied, while the update policy is used for end users who intend to modify the ciphertext. In other words, the data owner specifies two policies: one for decryption and another for modification. Fog nodes play a role in encryption by partially encrypting the data according to access policy, while the data owner completes the encryption phase with the access and update policies and sends it to cloud. IoT devices are limited resources that are connected to fog nodes and used by end users who would like to access the stored encrypted data in the cloud. To access data, the end user's attribute set must satisfy the access policy in ciphertext. Then, the fog node plays a second role to partially decrypt the ciphertext and let the user perform the rest of the decryption to recover the data. Once the user obtains access to the data, he or she might wish to modify and re-encrypt it. A signature-based attribute is applied, and a fog node plays a third role by supporting the request of the user to update the ciphertext. The partial signature is created by a fog node and is used to generate the user's signature. The CSP verifies the signature from end users and renews the ciphertext if the user's attribute set satisfies the update policy defined by the data owner.

The authors of [41] proposed a Chosen-Ciphertext Attack (CCA) security model for ABE with outsourced decryption in fog computing. The CCA is an attack in which the cryptanalyst can collect information by obtaining the decryptions of the chosen ciphertext. Once the information is gathered, the adversary tries to use the collected information to retrieve the user's secret key, which is used to decrypt the ciphertext. This model outsources the decryption operation to fog devices and consists of six algorithms: key generation, key extraction, outsourced decryption key generation, encryption, outsourced decryption by fog devices, and decryption by IoT devices. Two formats of the ciphertext are presented: the original ciphertext, which is generated from an encryption algorithm, and transformed ciphertext, which is executed by the outsourced decryption in fog devices. The paper presented two cases in which the attacker might try to figure out the ciphertext: one with the original ciphertext, and the other with the transformed ciphertext. Each method has several phases that are explained in detail in the paper. One way to detect a CCA is to check the validity of the ciphertext. Since two decryptions are needed in an Outsourced Decryption-Attribute-Based encryption scheme, ciphertext transformation by proxy and transformed ciphertext decryption by the decryptor, two techniques are used. These techniques are (1) Asymmetric and Symmetric Encryption Schemes, proposed in [48], and (2) Identity-Based Encryption, proposed in [49]. For Identity-Based Encryption techniques, which support verifiability, the proxy checks the validity before transformation. The decryptor can also check the validity of the transformed ciphertext using the Asymmetric and Symmetric Encryption technique. Therefore, the two techniques (Asymmetric and Symmetric Encryption and Identity-Based Encryption) are applied on an ABE scheme. The authors showed the cost of the algorithms (KeyGen, Ext, OKGen, Enc, and TDec) with a collection of attributes. The proposed scheme supported outsourced decryption in which the heavy computation is outsourced to fog devices, as the IoT devices have limited resources. However, the scheme does not support outsourced encryption, ciphertext update, and attribute revocation. When the number of data owners is increased, it is difficult to compute the encryption operations on the limited IoT devices without supporting the outsourced encryption to release the computational overhead from IoT devices.

The authors in [42] proposed a protocol of encrypted key exchange based on CP-ABE to secure the communications between fog nodes. In this protocol, communications between fog nodes and the cloud are confidential. The system model consists of several entities: a cloud, a key generator server, fog nodes, and IoT devices. The cloud is responsible for defining the access structure and executing the encryption to produce the ciphertext. Fog nodes are deployed on the network and each one is associated with a set of attributes that are defined by the access policy in the ciphertext. If the fog node's attribute set satisfies the access policy defined by the cloud, the fog node can decrypt the ciphertext and obtain the shared key.

In [11], Sun *et al.* proposed an attribute-based searchable encryption scheme based on cloud-fog computing. The proposed framework integrated the ABE technology and searchable encryption technology to achieve search-based keywords with fine-grained AC simultaneously. The CP-ABE with multiple authorities was also proposed to manage attribute creation and secret key distribution. The scheme has six entities: central authorities, attribute authority, CSP, fog nodes, the data owner, and the end user. In their scheme, due to the limited resources available by end users and data owners, part of the encryption and decryption operations are outsourced to the attached fog nodes. Therefore, the high computation overhead on end users and data owners are reduced. Personal health records in hospitals are an example of an application of the proposed scheme. One of the limitations is that the keyword sets are taken from the actual encrypted file in the cloud, which introduces the possibility of a chosen-ciphertext attack.

In [9], the authors proposed a keyword search over encrypted data system in fog computing that supported a fine-grained AC using CP-ABE. The system also supported attribute updates by updating the user's secret key and attributes associated with the ciphertext. In addition, they provided a multiple keywords option in a single search query,

which locates the data quickly and reduces the range of retrieved data. The system has five entities: a key generator center, a data owner, CSP, an end user, and a fog node. The system supported outsourcing encryption and decryption by moving part of the computational overhead, including file encryption and decryption, from the data owner or end user to the chosen fog nodes. They presented a security analysis that prevented two types of attacks: Chosen Keyword Attack and Chosen Plaintext Attack. One of the limitations is that, when the number of fog nodes and end users increases, the single key generator center is not enough to manage the distribution of secret keys and the creation of attributes for fog nodes and end users.

V. SECURITY AND PRIVACY ISSUES RELATED TO ACCESS CONTROL IN FOG COMPUTING

There are many security issues related to stored data in fog devices [50]. The ability to access and modify the user's data should only be permitted to authorized entities. Security and privacy requirements for several data services in fog computing, such as storage, sharing, and computation, are mentioned in [25]. AC ensures that only valid users are permitted to read, update, and/or send data within the fog. Thus, AC is used to prevent unauthorized access to data of any kind. Since fog computing is an extension of cloud computing, the security and privacy issues are inherited. Security and privacy issues that are relevant to AC in fog computing include:

A. TRUST IN FOG NODES

Since end users attach to the nearest fog node for real-time processing of their data, the trust level should be measured by the fog node or IoT device layer [51], [52]. Trust between fog nodes and IoT devices is important. The fog node that provides a service to the end user's device should confirm the authentication of the device. The end user's device that requests a service from a fog node should also be able to confirm the authentication of the intended fog node. AC models can be applied to measure the trust level when designing a trust model in the fog computing environment. The challenge will be how to define the trust level in fog computing [50], [53]. To measure the trust level of fog nodes, several attributes of fog nodes can be defined. One of the AC models is ABE, which provides fine-grained AC. The two types of attribute-based AC are CP-ABE and KP-ABE. Another challenge will be to determine who can verify the trust level of a fog node. The trusted authority can be defined to design a trust model in the fog computing environment. One of the roles of the attribute authority is to create a secret key that reflects fog or user attributes and manages these attributes.

B. DATA COMPUTATION IN FOG NODES

End users can offload their data to the nearest fog node for computation. However, outsourcing data to the fog node can cause data breaches. For example, in a smart grid, the reading of the smart meter by a fog node can leak household data [53]. The proposed solution to prevent data breaches is to apply AC models such as CP-ABE when outsourcing the data to the nearest fog nodes to achieve fine-grained AC. Several schemes involving the outsourcing of end users' data to fog nodes have been proposed in the literature. Collectively, these schemes suggest that the data should first be encrypted before offloading it to the fog nodes. The fog node can then perform part of the encryption and decryption of the data to relieve the heavy computation from a wearable end user device. Another important service is search over encrypted data. Serval schemes use a searchable encryption technology to search over encrypted data. Search-based keyword schemes that extend to achieve fine-grained AC using CP-ABE have been proposed in the literature.

C. ROGUE FOG NODE

A rogue fog node is a node that is reached by a malicious user. It appears as a legitimate fog node to other fog nodes in the network. Thus, a rogue fog node encourages other fog nodes to connect to it, which causes data damage or false data in the fog layer. One of the features of fog computing is to provide reliability in the fog node layer; however, a rogue fog node can lead to an attack to end users' data. Fog nodes must be protected against a malicious fog node when the end user sends sensitive information to it. When a fog node is divided and sends the computation task to several other fog nodes in the network, if one of fog nodes is a rogue node, it injects false data to the other fog nodes. Therefore, the security and privacy of end user will be destroyed [53]. An ABE scheme, which is a type of attribute-based AC, can be applied to provide confidence in end users and fog nodes.

D. FOG NODE PRIVACY

In fog computing, sensitive data can be disclosed because the fog node is closed to end users. When an end user offloads its task to the nearest fog node in the network, the location of the end user can be disclosed since location awareness is one of the features of fog computing. If an end user outsources its data to the nearest fog node, the fog node can indicate that the end user is close to that fog node. Once an end user outsources its data to several fog nodes in the network, the privacy of the user's location will be at risk [53]. AC can be a solution to address the issue of user privacy preservation and the security of fog nodes.

In a data breach, the user's information is disclosed and accessed by unauthorized users. When a fog node is performing its task of collecting computing end users' data, data breaches can occur on the end users' side or the fog side. Therefore, AC is needed to maintain the confidentiality of end users' data. One scenario that could occur is one of the fog nodes being reached by a malicious user, then acting as a legitimate fog node to the other fog nodes in the network. The malicious data from the attacker will then be delivered to the other fog devices in the network [54]. To solve the issue, there is a need for security mechanisms such as AC to protect fog devices and prevent malicious activities [18].

Another scenario that could put end users' privacy at risk is a fog node leaving the fog network permanently. In this case, entities interacting with the fog node that has left the network, such as the data owner, end user, or CSP, need to update their AC lists to avoid leftover access to a node that does not exist. AC should be applied to allow only authorized users to access the data—protecting the stored data in fog devices from unauthorized users is another challenge. It is also challenging to design a fine-grained AC system that supports scalability. When an end user's attributes are revoked, updating the user's attributes turns out to be challenging as the number of users increases.

E. PRIVACY PRESERVING IN FOG NODES

Fog computing, like any other computing model, is not immune to privacy issues, including those involving data privacy and location privacy [55], [56]. To solve the latency issue, fog nodes are close to the IoT devices, which facilitates the real-time processing capability in fog nodes. However, data can be overexposed and revealed to the outside world because of this additional fog layer. The data generated by IoT devices will be computed by the nearest fog nodes, and false data injection attacks can occur when data are outsourced to the fog nodes [57]. Since fog nodes are close to IoT devices, location privacy is another issue. When IoT devices subscribe to a specific fog node for processing demand, it can be inferred that the subscribed IoT devices are close to that fog node and far away from other fog nodes [55], [56]. Therefore, a privacy-preserving guarantee must be achieved in fog computing. AC models can be a solution to address the issue of privacy preserving. Since some AC models, such as ABE, are encryption-based, the promise of data confidentiality can be met. In addition, fine-grained AC can help in limiting the access to data. This can also ensure preserving the privacy of data and users [47].

VI. DISCUSSION & RESEARCH GAPS

A. DISCUSSION OF SEVERAL FEATURES IN DIFFERENT SCHEMES

There are several features that should be taken into consideration when designing AC models. These features are crucial to make the designed scheme more efficient and secure. The features are: (1) outsourcing encryption, (2) outsourcing decryption, (3) multiple authorities, (4) supportability in the fog computing environment, and (5) providing search-based keywords. Using AC models require several operations of encryption and decryption, which increases computation overhead. One of the desired features is to outsource part of the encryption and decryption when designing AC models. Therefore, to reduce the computation overhead, IoT devices can perform part of the encryption and decryption operations, with fog nodes performing the rest. There are several entities that interact with each other in AC models. One of these entities is the key authority, which is responsible for creating secret keys and distributing attributes to users. Thus, when the number of IoT devices increases, one key authority will not be enough to generate secret keys and distribute attributes to users. Designing AC models with multiple key authorities can significantly reduce the network congestion and improve the system's efficiency. In addition, multiple authorities can relieve the enormous effort required for the data owner to handle the user attributes. Designing AC models to be used in the context of fog computing can decrease the latency issues of IoT devices and relieve the computation overhead of encryption and decryption operations from IoT devices. Since end users should satisfy the access policies of the ciphertext, fog nodes can compare and execute part of the decryption in a timely manner before outsourcing decryption to the end user.

Using searchable encryption (SE) technologies is another feature that could decrease the range of retrieved data from the cloud. As the amount of data increases, integrating SEs in AC models becomes desirable. Therefore, combining the features mentioned can crucially improve the design of AC models. These features are important to researchers due to their benefits in many domains, such as medical care. Designing AC models in medical care is challenging since the privacy and security of patients' data are extremely important.

Table 2 compares existing works that propose AC schemes according to the features they have (i.e., outsourcing encryption, outsourcing decryption, multiple authorities, supportability in a fog computing environment, and providing search-based keywords). All schemes in Table 2 use AC strategies that require heavy computation due to the operations of encryption and decryption. Schemes described in [9]–[11], [34], [38], [39], [43], [45]–[47] support outsourcing encryption and decryption, and schemes in [33], [40], [41], [44] support only outsourcing the decryption operations. Outsourcing the heavy operations of encryption and decryption means that the computation overhead of the end users will be decreased. However, some of the schemes solve the latency issue, while other do not. Schemes in [9]–[11], [34], [45]–[47] provide outsourcing of encryption and decryption and solve the latency issue by introducing a fog computing layer between IoT devices and the cloud. The schemes presented in [38]–[40], [43], [44] offer outsourcing of encryption and decryption, except those in [40], [44], which only offer outsourcing of decryption. The latency for all of them can be high due to the use of serval cloud computing servers. Multiple authorities enhance the scalability of building a model and reduce the computation overhead on a single authority. Data owners and end users' attributes are distributed by the attribute authority. As the number of end users and data owners increases, a single attribute authority will not be enough to handle the distribution of users' attributes. Few schemes used multiple attribute authorities to improve AC in their proposed schemes. Authors of [11], [31], [33], [37], [45]–[47] developed a scheme that is highly scalable by introducing several distributing attribute authorities in their work. Schemes in [31], [33], [37], [45]–[47] support only multiple attribute authorities and AC. The scheme presented in [11] supports

multiple attribute authorities and all features in Table 2. SE technologies have been well studied in the literature. One of the known SE technologies is search-based keywords, which gained considerable attention in cloud computing for several years. Some schemes presented in Table 2 provided search-based keywords in the context of cloud computing so that the designed model has both SE technologies and fine-grained AC. Few papers enhanced their schemes by deploying CP-ABE and SE technologies in a fog computing layer that can solve the latency issue. In such schemes, encryption and decryption operations are outsourced to fog nodes and the computation overhead in end users' devices will be reduced. For example, one paper introduced a scheme that uses CP-ABE and search-based keywords and deployed it in the fog using multiple attribute authorities. Additionally, scheme [9] proposed a fine-grained keyword search with outsourcing encryption and decryption in fog computing, while scheme [11] presented ABE and keyword search for personal health records in fog computing using multiple attribute authorities.

B. GAPS AND FUTURE RESEARCH DIRECTIONS

Fog computing and cloud computing are similar in nature and highly coupled. However, solutions built to address the cloud efficiency, security, and privacy issues cannot necessarily be applied to fog computing. As previously discussed, multiple operations of AC models can have a major impact on fog computing solutions. This section outlines the research gaps that need to be addressed to enhance AC models in fog computing:

1) AUDITING MECHANISM

Designing an AC model with an auditing mechanism is necessary in distributed fog nodes. Auditing, in this case, is important to periodically check users' attributes and make sure that the attributes are valid, and that the users' privileges are not outdated. Moreover, in highly scalable environments, an auditing mechanism becomes important for the robustness of AC. Since cloud computing, IoT, and fog computing are highly coupled and scalable, there will be a massive number of joining and leaving nodes (e.g., fog, cloud, IoT). This mandates continuous and thorough auditing to maintain the confidentiality and integrity of the data and applications. The existence of nodes with more privileges than needed also increases the demand of computation power to handle operations like encryption and decryption, which negatively impacts the environment performance. Therefore, intelligent auditing mechanisms are needed to automatically search for policy violations and update access policies and users' attributes.

2) FINE-GRAINED ACCESS CONTROL

Proposing fine-grained AC is essential in widely distributed fog nodes. When a number of IoT devices connect to fog nodes, fog nodes apply one or more AC models to grant access to a number of authorized IoT devices. Fine-grained AC can be introduced in fog computing to limit the access to specific data, and each fog node can apply its own access policy for its own IoT devices. This is important, as it will give administrators more control and flexibility to securely and effectively overexpose and underexpose data. Designing fine-grained AC models for distributed fog nodes is necessary; thus far, however, little or no work has been done to tackle this challenge.

3) COVERING MORE FEATURES FOR BETTER EFFICIENCY

Designing and implementing AC models that cover more features is important for the efficiency of the model. Therefore, we surveyed AC models and their supporting features to better understand how these features work and how to integrate them in a future AC model. More AC features can still be explored and integrated to build efficient AC in fog computing.

VII. CONCLUSION

Fog computing is a new computing paradigm that provides real-time processing at the edge of the network, close to IoT devices. AC models can be applied in fog computing to preserve the privacy of IoT data and to protect the system and users' data. Several security and privacy issues in fog computing can be solved using one or more AC model(s); however, the choice of an AC model is dependent on the application's requirements. In this paper, we thoroughly discussed fog computing and AC models. Then, we presented the state of the art in the field of fog computing AC. We also discussed some security and privacy issues relevant to AC in fog computing. Several features that that are known to produce efficient AC models in fog computing were discussed and research gaps were outlined.

In our future research, we plan to propose an AC model that supports more features for better efficiency and security. We also plan to investigate designing a fine-grained AC model for fog computing-aided environments.

REFERENCES

[1] Z. Wan, "Cloud computing infrastructure for latency sensitive applications," in *Proc. IEEE 12th Int. Conf. Commun. Technol.*, Nanjing, China, Nov. 2010, pp. 1399–1402, doi: 10.1109/ICCT.2010.5689022.

[2] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in *Proc. ASE BigData SocialInformatics*, 2015, pp. 1–6.

[3] F. Y. Okay and S. Ozdemir, "A fog computing based smart grid model," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Yasmine Hammamet, Tunisia, May 2016, pp. 1–6, doi: 10.1109/ISNCC.2016.7746062.),

[4] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: A review of current applications and security solutions," *J. Cloud Comput.*, vol. 6, no. 1, p. 19, Dec. 2017, doi: 10.1186/s13677-017-0090-3.

[5] K. Lee, D. Kim, D. Ha, U. Rajput, and H. Oh, "On security and privacy issues of fog computing supported Internet of Things environment," in *Proc. 6th Int. Conf. Netw. Future (NOF)*, Montreal, QC, Canada, Sep. 2015, pp. 1–3, doi: 10.1109/NOF.2015.7333287.

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

[7] G. Rahman and C. C. Wen, "Fog computing, applications, security and challenges, review," *Int. J. Eng. Technol.*, vol. 7, no. 3, p. 1615, Jul. 2018, doi: 10.14419/ijet.v7i3.12612.

[8] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018, doi: 10.1109/ACCESS.2018.2866491.

[9] Y. Miao, J. Ma, X. Liu, J. Weng, H. Li, and H. Li, "Lightweight fine-grained search over encrypted data in fog computing," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 772–785, Sep. 2019, doi: 10.1109/TSC.2018.2823309.

[10] 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, doi: 10.1016/j.future.2016.12.015.

[11] J. Sun, X. Wang, S. Wang, and L. Ren, "A searchable personal health records framework with fine-grained access control in cloud-fog computing," *PLoS ONE*, vol. 13, no. 11, Nov. 2018, Art. no. e0207543, doi: 10.1371/journal.pone.0207543.

[12] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "IFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments," *Softw., Pract. Exper.*, vol. 47, no. 9, pp. 1275–1296, Sep. 2017, doi: 10.1002/spe.2509.

[13] Fog Computing. (2015). *Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are*. [Online]. Available: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf

[14] M. Iorga, L. Feldman, R. Barton, M. J. Martin, N. S. Goren, and C. Mahmoudi, "Fog computing conceptual model," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NIST SP 500-325, Mar. 2018.

[15] A. Abuhussein, H. Bedi, and S. Shiva, "Evaluating security and privacy in cloud computing services: A Stakeholder's perspective," in *Proc. Int. Conf. Internet Technol. Secured Trans.*, 2012, pp. 388–395.

[16] A. Abuhussein, F. Alsubaei, and S. Shiva, "Toward an effective requirement engineering approach for cloud applications," in *Software Engineering in the Era of Cloud Computing*, M. Ramachandran and Z. Mahmood, Eds. Cham, Switzerland: Springer, 2020, pp. 29–50.

[17] R. Kumar Naha, S. Garg, and A. Chan, "Fog computing architecture: Survey and challenges," 2018, *arXiv:1811.09047*. [Online]. Available: http://arxiv.org/abs/1811.09047

[18] P. Zhang, J. K. Liu, F. R. Yu, M. Sookhak, M. H. Au, and X. Luo, "A survey on access control in fog computing," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 144–149, Feb. 2018, doi: 10.1109/MCOM.2018.1700333.

[19] M. Aazam and E.-N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 2015, pp. 687–694, doi: 10.1109/AINA.2015.254.

[20] J. Dizdarević, F. Carpio, A. Jukan, and X. Masip-Bruin, "A survey of communication protocols for Internet of Things and related challenges of fog and cloud computing integration," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–29, Feb. 2019, doi: 10.1145/3292674.

[21] F. Alsubaei, A. Abuhussein, and S. Shiva, "An overview of enabling technologies for the Internet of Things," in *Internet Things A to Z*. Hoboken, NJ, USA: Wiley, 2018, pp. 77–112.

[22] Q. Wang, D. Chen, N. Zhang, Z. Ding, and Z. Qin, "PCP: A privacy-preserving content-based publish–subscribe scheme with differential privacy in fog computing," *IEEE Access*, vol. 5, pp. 17962–17974, 2017, doi: 10.1109/ACCESS.2017.2748956.

[23] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb)*, Washington, DC, USA, Nov. 2015, pp. 73–78, doi: 10.1109/HotWeb.2015.22.

[24] A. Ahmed, H. Arkian, D. Battulga, A. J. Fahs, M. Farhadi, D. Giouroukis, A. Gougeon, F. O. Gutierrez, G. Pierre, P. R. Souza, Jr., M. A. Tamiru, and L. Wu, "Fog computing applications: Taxonomy and requirements," 2019, *arXiv:1907.11621*. [Online]. Available: http://arxiv.org/abs/1907.11621

[25] Y. Guan, J. Shao, G. Wei, and M. Xie, "Data security and privacy in fog computing," *IEEE Netw.*, vol. 32, no. 5, pp. 106–111, Sep. 2018, doi: 10.1109/MNET.2018.1700250.

[26] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: Architecture, key technologies, applications and open issues," *J. Netw. Comput. Appl.*, vol. 98, pp. 27–42, Nov. 2017, doi: 10.1016/j.jnca.2017.09.002.

[27] *Health Information Privacy | HHS.gov*. Accessed: Mar. 14, 2020. [Online]. Available: https://www.hhs.gov/hipaa/index.html

[28] W. Stallings and L. Brown, *Computer Security: Principles and Practice*. Harlow, U.K.: Pearson, 2019. Accessed: Jan. 28, 2020. [Online]. Available: https://content/one-dot-com/one-dot-com/us/en/higher-education/program.html

[29] M. Mammass and F. Ghadi, "Access control models: State of the art and comparative study," in *Proc. 2nd World Conf. Complex Syst. (WCCS)*, Agadir, Morocco, Nov. 2014, pp. 431–435, doi: 10.1109/ICoCS.2014.7060973.

[30] N. Smyth, *Security+Essentials*. Raleigh, NC, USA: Payload Media, 2010.

[31] K. Yang, X. Jia, K. Ren, and B. Zhang, "DAC-MACS: Effective data access control for multi-authority cloud storage systems," in *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 2013, pp. 2895–2903, doi: 10.1109/INFCOM.2013.6567100.

[32] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of Everything*, B. Di Martino, K.-C. Li, L. T. Yang, and A. Esposito, Eds. Singapore: Springer, 2018, pp. 103–130.

[33] K. Vohra and M. Dave, "Multi-authority attribute based data access control in fog computing," *Procedia Comput. Sci.*, vol. 132, pp. 1449–1457, 2018, doi: 10.1016/j.procs.2018.05.078.

[34] Q. Huang, Y. Yang, and L. Wang, "Secure data access control with ciphertext update and computation outsourcing in fog computing for Internet of Things," *IEEE Access*, vol. 5, pp. 12941–12950, 2017, doi: 10.1109/ACCESS.2017.2727054.

[35] F. Alsubaei, A. Abuhussein, and S. Shiva, "Ontology-based security recommendation for the Internet of medical things," *IEEE Access*, vol. 7, pp. 48948–48960, 2019, doi: 10.1109/ACCESS.2019.2910087.

[36] A. Abuhussein, S. Shiva, and F. T. Sheldon, "CSSR: Cloud services security recommender," in *Proc. IEEE World Congr. Services (SERVICES)*, Jun. 2016, pp. 48–55, doi: 10.1109/SERVICES.2016.13.

[37] 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 Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 131–143, Jan. 2013, doi: 10.1109/TPDS.2012.97.

[38] Z. Zhou and D. Huang, "Efficient and secure data storage operations for mobile cloud computing," in *Proc. 8th Int. Conf. Netw. Service Manage. (CNSM) Workshop Syst. Virtualiztion Manage. (SVM)*, Oct. 2012, pp. 37–45.

[39] M. R. Asim, M. Petkovic, and T. Ignatenko, "Attribute-based encryption with encryption and decryption outsourcing," in *Proc. 12th Austral. Inf. Secur. Manage. Conf.*, 2014, pp. 21–28, doi: 10.4225/75/57b65cc3343d0.

[40] X. Mao, J. Lai, Q. Mei, K. Chen, and J. Weng, "Generic and efficient constructions of attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Depend. Sec. Comput.*, vol. 13, no. 5, pp. 533–546, Sep. 2016, doi: 10.1109/TDSC.2015.2423669.

[41] C. Zuo, J. Shao, G. Wei, M. Xie, and M. Ji, "CCA-secure ABE with outsourced decryption for fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 730–738, Jan. 2018, doi: 10.1016/j.future.2016.10.028.

[42] A. Alrawais, A. Alhothaily, C. Hu, X. Xing, and X. Cheng, "An attribute-based encryption scheme to secure fog communications," *IEEE Access*, vol. 5, pp. 9131–9138, 2017, doi: 10.1109/ACCESS.2017.2705076.

[43] H. Wang, D. He, J. Shen, Z. Zheng, C. Zhao, and M. Zhao, "Verifiable outsourced ciphertext-policy attribute-based encryption in cloud computing," *Soft Comput.*, vol. 21, no. 24, pp. 7325–7335, Dec. 2017, doi: 10.1007/s00500-016-2271-2.

[44] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: Outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 715–725, Sep. 2017, doi: 10.1109/TSC.2016.2542813.

[45] K. Fan, J. Wang, X. Wang, H. Li, and Y. Yang, "A secure and verifiable outsourced access control scheme in fog-cloud computing," *Sensors*, vol. 17, no. 7, p. 1695, Jul. 2017, doi: 10.3390/s17071695.

[46] Q. Xu, C. Tan, Z. Fan, W. Zhu, Y. Xiao, and F. Cheng, "Secure data access control for fog computing based on multi-authority attribute-based signcryption with computation outsourcing and attribute revocation," *Sensors*, vol. 18, no. 5, p. 1609, May 2018, doi: 10.3390/s18051609.

[47] K. Xue, J. Hong, Y. Ma, D. S. L. Wei, P. Hong, and N. Yu, "Fog-aided verifiable privacy preserving access control for latency-sensitive data sharing in vehicular cloud computing," *IEEE Netw.*, vol. 32, no. 3, pp. 7–13, May 2018, doi: 10.1109/MNET.2018.1700341.

[48] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," *J. Cryptol.*, vol. 26, no. 1, pp. 80–101, Jan. 2013, doi: 10.1007/s00145-011-9114-1.

[49] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Proc. Int. Conf. Theory Appl. Cryptographic. Techn.*, vol. 3027. Berlin, Germany: Springer, 2004, p. 16, doi: 10.1007/978-3-540-24676-3_13.

[50] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, and V. Kumar, "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, pp. 19293–19304, 2017, doi: 10.1109/ACCESS.2017.2749422.

[51] F. Alsubaei, A. Abuhussein, and S. Shiva, "Security and privacy in the Internet of medical things: Taxonomy and risk assessment," in *Proc. IEEE 42nd Conf. Local Comput. Netw. Workshops (LCN Workshops)*, Oct. 2017, pp. 112–120, doi: 10.1109/LCN.Workshops.2017.72.

[52] F. Alsubaei, A. Abuhussein, V. Shandilya, and S. Shiva, "IoMT-SAF: Internet of medical things security assessment framework," *Internet Things*, vol. 8, Dec. 2019, Art. no. 100123, doi: 10.1016/j.iot.2019.100123.

[53] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.*, 2015, pp. 685–695.

[54] 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, Jul. 2016, doi: 10.1002/cpe.3485.

[55] N. Abubaker, L. Dervishi, and E. Ayday, "Privacy-preserving fog computing paradigm," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2017, pp. 502–509, doi: 10.1109/CNS.2017.8228709.

[56] M. A. Ferrag, A. Derhab, L. Maglaras, M. Mukherjee, and H. Janicke, "Privacy-preserving schemes for fog-based IoT applications: Threat models, solutions, and challenges," in *Proc. Int. Conf. Smart Commun. Netw. Technol. (SaCoNeT)*, Oct. 2018, pp. 37–42, doi: 10.1109/SaCoNeT.2018.8585538.

[57] Y. Zhang, J. Zhao, D. Zheng, K. Deng, F. Ren, X. Zheng, and J. Shu, "Privacy-preserving data aggregation against false data injection attacks in fog computing," *Sensors*, vol. 18, no. 8, p. 2659, Aug. 2018, doi: 10.3390/s18082659.

MOHAMMED A. ALEISA (Member, IEEE) received the B.S. degree in computer science from King Abdulaziz University and the M.S. degree in computer science from The University of Colorado, Denver, USA. He is currently pursuing the Ph.D. degree in computer science with The University of Idaho, Moscow, USA. He has been a Lecturer with Majmaah University, Saudi Arabia, since 2011, and is currently on sabbatical leave. His research interests include security and privacy in fog computing and cloud computing.

ABDULLAH ABUHUSSEIN (Member, IEEE) received the B.Sc. degree in computer science, in 1999, the M.Sc. degree in information systems management from Ferris State University, in 2002, and the Ph.D. degree in computer science from The University of Memphis. In 2017, he joined the Department of Information Systems, Herberger Business School, St. Cloud State University, as an Assistant Professor. In his Ph.D. research, he focused on pragmatic cloud security assessment framework and stakeholder's perspective in cybersecurity. He is teaching courses on security, software engineering, and cloud computing as a Lecturer with various educational institutions. His research interests include cloud computing, cloud security, security economics, the Internet of Things (IoT), software engineering, security and privacy, and security metrics. He holds a number of refereed publications in related venues, presented articles in various conferences, and served as a reviewer for some journals, including the IEEE TRANSACTIONS ON CLOUD COMPUTING.

FREDERICK T. SHELDON (Senior Member, IEEE) received the bachelor's degree in computer science from the University of Minnesota Twin Cities and the master's degree from the University of Texas at Arlington. He has 35+ years in the fields of software engineering and computer science (and security) engaged as an Engineer, a Principal Investigator, a Research Scientist, a Business Developer, a Faculty Member, and an Administrator. He has held faculty appointments at The University of Colorado Colorado Springs, Washington State University, the University of Memphis, and Wuhan University. He is currently a Professor with the University of Idaho Coeur d'Alene. He has held research and development positions at the Oak Ridge National Laboratory (ORNL) and three fortune 100 companies (GD, LMCO, and Raytheon/TI), including a Fellowship and National Academy Postdoctoral at NASA Langley and a Visiting Scholar at NASA Ames and Stanford University. He has coauthored 160+ articles, 12 editorships, four U.S. patents, and chaired, facilitated, and participated in numerous national research and development venues, including an invited speaker, a panelist, and a moderator. His research interests for the last 15 years are focused on cyber and information security. He enjoys teaching upper division and graduate level courses. He is a Senior Member of ACM. He received the Sigma Xi Research and UT-Battelle Key Contributor and Significant Event Awards at ORNL.