Security and Privacy for Internet of Things and Fog Computing

Carlos Alejandro Peñuelas Angulo, Claudia Feregrino Uribe

Reporte Técnico No. CCC-21-003

23 de agosto de 2021

Luis Enrique Erro 1

Sta. Ma. Tonantzintla

72840, Puebla, México

Security and Privacy for Internet of Things and Fog Computing

Carlos Alejandro Peñuelas Angulo Claudia Feregrino Uribe

Computer Science Department
National Institute of Astrophysics, Optics and Electronics
Luis Enrique Erro # 1, Santa María Tonantzintla, Puebla, 72840, México
E-mail: {alejandrop, cferegrino}@inaoep.mx

Abstract

Internet of Things (IoT) applications have been widely extended in recent years to improve people's quality of life. A significant part of them require the collection of data that could potentially reveal information of the original owner and it is commonly stored in an untrusted cloud service provider (CSP). To protect this information, the standard solution is to apply an access control model over the outsourced data. However, the conventional approaches, such as data encryption and key management performed by the CSP, raises several security and privacy concerns since the data owner is no longer in direct control of her/his data. For these reasons, fine-grained access control schemes based on attribute based cryptography for data outsourced to the cloud have been proposed. However, existing proposals are still limited in functionality and are not flexible enough to support the dynamic nature of the IoT environment. For these reasons, we propose a fine-grained attribute-based access control scheme for applications in the Fog-IoT environment that includes management mechanisms for user revocation, attribute update, and access policy update. In addition, we propose the additional feature of privacy-preserving data aggregation to further improve the control over shared data and reduce the communication overhead, by only sharing aggregated data to specific entities. In this proposal, we consider fog computing as an interesting paradigm that can help the resource-constrained IoT devices to perform expensive operations of attribute-based cryptography.

Keywords: *Access Control, Attribute-based Cryptography, Fog-enabled IoT, Privacy-preserving Data Aggregation.*

Contents

1 Introduction

Internet of Things (IoT) is the concept of connecting sensors and objects to the internet to enable several applications based on the collection and processing of data along with communication between different objects [1]. The most accepted model of IoT relies on the outsourcing of collected data to a cloud service provider (CSP) where data is managed and processed. Furthermore, there is a wide range of applications based on IoT dealing with highly important or sensitive data: smart homes/buildings, industry 4.0, healthcare applications, among others [2]. For this reason, it is crucial to implement security and privacy mechanisms for the IoT environment.

A promising measure to protect data outsourced to the cloud is the implementation of fine-grained access control [3]. Existing techniques have explored the use of attribute based cryptography algorithms to design secure access control schemes in this context. However, IoT devices commonly lack of computation or storage resources to afford the required processing for that task. For this reason, novel computing paradigms such as fog computing (FC) have emerged with the aim of reducing the problems related to computation, communication, storage, and latency in IoT applications enabled by the cloud. In FC, there is an intermediate layer, called fog layer, between the end devices and the CSP that brings cloud capabilities to the edge of the network.

In this document, motivated by IoT applications that require security and privacy for outsourced data, we present a research proposal that focuses on the development of a attribute-based access control scheme for the Fog-IoT environment. We also consider the scalability of the environment an other functionalities such as management mechanisms, and privacy-preserving data aggregation.

2 Background

2.1 Internet of things

The internet of things (IoT) is an interconnection of objects or *things* with access to the internet that allows exchanging data and information to enable different applications [2]. These objects can be sensors, meters, smart phones, vehicles, actuators and other devices with enough hardware and software capabilities to obtain and/or produce data and perform communications with other elements. The IoT emerged to enable applications that improve the quality of life of people including smart home/buildings, smart grids, smart cities, industry 4.0, smart healthcare, among others.

The most common model used in current deployments of IoT is based on the collection of data from the end users or end entities (vehicles, buildings, facilities, etc.) and the use of cloud-based service providers, i.e., cloud computing. This model maintains a highly centralized structure that, considering the exponential growth of the number of IoT devices and users [4], deals with serious problems such as communication delays, processing of a large amount of data, traffic channel congestion, and increasing communication costs [5]. In this situation, novel network architectures have been proposed to overcome the mentioned drawbacks in the cloud-based model of IoT. One of them is fog computing, which has emerged as a novel computing paradigm that brings cloud capabilities closer to the edge of the network. By doing so, FC has become an alternative for those applications that require location awareness, real-time response, and reduction of

communication cost. Additionally, fog nodes can collaborate with the end user devices that are resource-constrained to reduce their processing burden [2].

2.2 Fog computing

The term *fog computing* is defined by Cisco [6] as an extension of the cloud to the edge of the network that provides different services such as storage, processing, and networking services between the cloud and the end devices. Among the main characteristics of this paradigm are low latency, location awareness, mobility support, geographical distribution, decentralization, heterogeneity, and low bandwidth cost [6, 7]. These characteristics of fog computing reveal that, despite that the fog is an extension of the cloud, this extension is non-trivial.

A widely used architecture for fog computing is the three-tier architecture [8] that is illustrated in Fig. 1. The first tier is composed of the IoT or end devices (smart watches, smart meters, building sensors, vehicles, etc.). The second tier is also known as the fog computing layer where the fog nodes (smart phones, security cameras, laptops, gateways, etc.) are able to share processing and storage capabilities [5]. The third tier is the cloud computing layer which is assumed to posses high computing and storage resources [5]. In addition to the three tiers, some interfaces between layers and nodes are considered in this architecture. As shown in Fig. 1, the fog nodes are deployed between the cloud and the IoT layers and are considered in the three types of interfaces: Cloud-Fog, Fog-Fog, and Fog-IoT. These special features in addition to the intrinsic characteristics of fog computing are the key for the improvement of IoT applications compared to the cloud-based IoT.

However, given the specific properties of fog computing and IoT environment (Fog-IoT), the security and privacy issues that arise in this integration are also specific and, due to the non-triviality of its definition, the existing solutions for the security and privacy of the cloud are not suitable for the Fog-IoT environment [9].

2.3 Access control

A large number of IoT applications outsource the collected data to a cloud service provider (CSP) where a data sharing mechanism can be implemented. This model brings serious security concerns since the data owner (DO) no longer has direct control over the stored data, and the cloud environment is considered a not fully trusted entity [3]. A curious CSP or a malicious user may try to access the stored data and use it to obtain sensitive information about the data or even the DO.

On the CSP side, an usual measure is to apply symmetric encryption algorithms to protect the outsourced data. This mechanism works as follows: data owners send data they generate to the cloud in plaintext where the CSP, in addition to data storage, is in charge of the key generation, encryption, decryption, and key management processes. Hence, data owners must trust the CSP to provide these services, which has been mentioned as one of the main concerns in the cloud environment [3].

Access control is a measure to provide data confidentiality and defining mechanisms for data access. In the Fog-IoT environment, access control is an essential requirement. Data stored in the cloud can be requested by several users in different locations, and these data must be retrieved only by authorized users or devices [10].

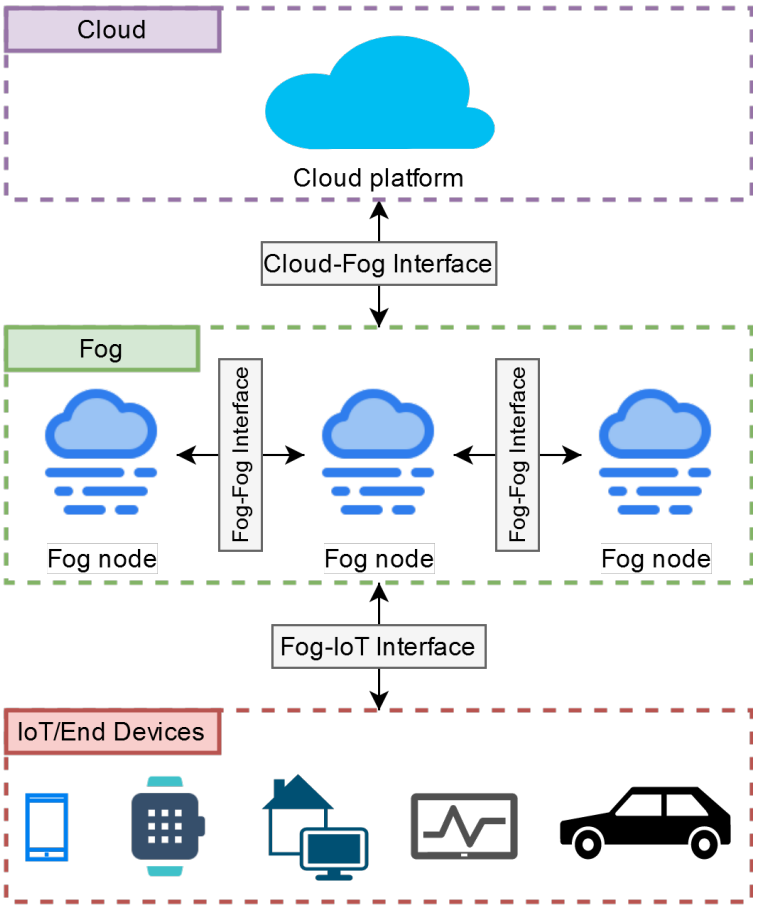

Figure 1. Three-tier fog computing architecture [5].

There are several access control models in the literature with distinct properties that are suitable for different applications and scenarios. Figure 2 shows four of the most extended access control models [10, 11] which are described below.

1. **Discretionary access control (DAC):** It allows the data owner to decide the access rights for every user in the system. These rights are established with regard to the identity of the data users. This model is highly flexible since the access rights may be different for each participant according to the decisions of the data owner. However, it is considered potentially insecure [10]. Hence, it is mostly used in applications that require certain compatibility with older systems which may produce high management overhead in fog environment.

2. **Mandatory access control (MAC):** This model is based on the idea of assigning security labels to resources and identifying users with a security level. This allows to establish a multi-layer security mechanism. With MAC, when a user requests a resource, the security labels of the resource are compared with the security level of the user. If both match, the access is granted [11].

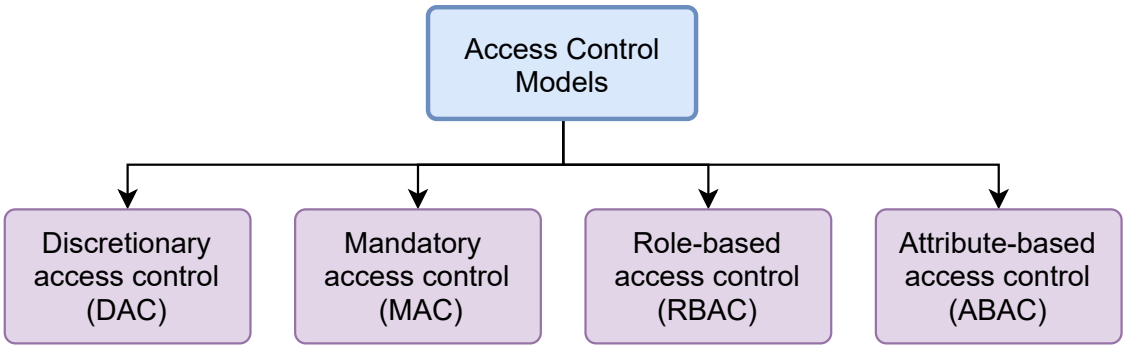

Figure 2. Access control models [10].

3. **Role-based access control (RBAC):** This model is motivated by the idea that the responsibility of a subject has more impact on the system than the subject itself [10]. In RBAC model, the access rights are defined based on the roles of the users in the system. These roles may include, for instance, a researcher, a professor, or a student in a university. Hence, RBAC allows fine-grained access control, it is more scalable than other models (DAC and MAC). For these reasons, this model is more suitable for the fog computing environment [10].

4. **Attribute-based access control (ABAC):** This model is based on the use of attributes to define the access rights of the data users. In ABAC, a universe of attributes is defined and the system administrator assigns subsets of these attributes to each involved party that eventually may request for resources. The model allows the data owner to directly establish access policies attached to the resources. These policies indicate the attributes that must be owned by a data user in order to access to the resources. In this sense, the ABAC model allows to define a fine-grained access control mechanism in such a way that the data owner has a high level of control over delegated resources [10]. In addition, this model does not have a high dependency on the system administrator to define the access structures. Therefore, the features of the ABAC model are highly suitable for applications in the fog computing setting.

2.4 Attribute-based cryptography

In general, public-key cryptography requires the management of several key pairs linked to individual entities. Each entity has an identity associated with a public key, fully available for all the authorized participants, and a private key, which only the individual entity should know. In this setting, an authorized party P_1 can send encrypted messages to a second party P_2 by retrieving the public key of P_2 and using an asymmetric encryption algorithm. Then, P_2 can use its private key and the corresponding asymmetric decryption algorithm to recover the messages. Such a system is an example of one-to-one cryptography and, for several applications, more flexible cryptographic mechanisms are required.

Attribute-based Cryptography (ABC) comprises a set of cryptographic techniques that enable the concept of one-to-many cryptography. Unlike public-key cryptography, in ABC there exists

a universe of attributes. Subsets of these attributes can be used to define groups of parties with different privileges or access rights. For instance, an ABC algorithm for data encryption can encrypt a message with regard to a set of attributes or an access formula. Then, the message can be decrypted only by those entities that meet the required attributes in the formula. For this reason, ABC has been widely used to develop ABAC schemes [11, 12].

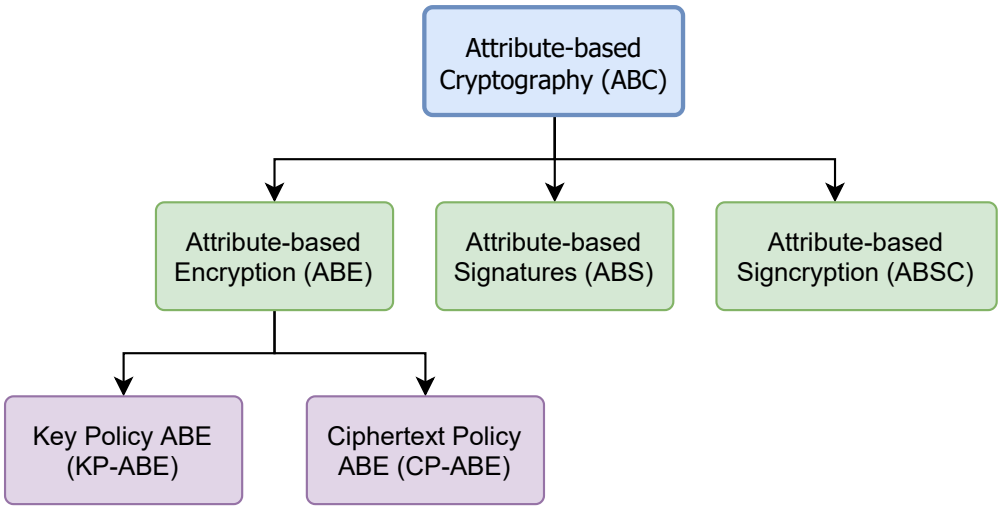

Figure 3. Classification of attribute based cryptography techniques.

Figure 3 shows a classification of different ABC techniques: attribute-based encryption, attribute-based signatures, and attribute-based signcryption. A brief description of them is shown below:

1. **Attribute-based encryption (ABE):** It was first introduced by Sahai and Waters [13] as an extension of identity-based encryption (IBE) [14, 15], another interesting cryptographic primitive that generalizes public-key cryptography. In an ABE scheme, for instance, the identity of a user may be linked to a number of attributes and a set of secret keys related to those attributes is issued to the user. To encrypt a message, an access policy is defined according to a subset of the attribute universe. Then, the message is decrypted only by the users whose attributes, and therefore their secret keys, can fulfill the access policy. In this sense, ABE enables the function of one-to-many encryption because the selected attributes may be owned by different users without having the same secret keys. Finally, there exist two modalities of ABE depending on whether the access policies are associated with the secret keys or the ciphertext: Key-policy ABE and Ciphertext-policy ABE [16], respectively. The example described previously is an example of a CP-ABE scheme.

    (a) **Key-policy ABE (KP-ABE):** In this class of ABE, the encryption is performed using a set of attributes that are attached to the message while the access policies are linked to users' secret key [17]. Thus, the decryption of a message requires a match between the attributes assigned to the ciphertext and the access policy embedded into the users' secret keys.

(b) **Ciphertext-policy ABE (CP-ABE):** Unlike KP-ABE, in CP-ABE the attributes are associated to the users themselves and their secret keys while the access policies are linked to encrypted data. During the encryption process, an access policy is first defined and the ciphertext is produced according to that policy. Only the users whose attributes satisfy the access policy of a given ciphertext are allowed to decrypt the message. Hence, CP-ABE seems to be more suitable for fine-grained access control schemes than KP-ABE because, with the given definition of KP-ABE, the data owner does not have complete control over the potential recipients of the message. An example of a CP-ABE scheme is shown in Figure 4.

2. **Attribute-based signature (ABS):** Attribute-based signatures were introduced by Maji *et al.* in [18]. ABS is a signature algorithm to produce authenticated messages using attributes instead of identities. Similarly to ABE, in an ABS scheme users are associated with attributes that use their secret parameters to sign data, and verification process only reflects that the signer posses some attributes without the revealing identity of this user.

3. **Attribute-based signcryption (ABSC):** It is a signcryption primitive [19], i.e., an algorithm that allows to sing and encrypt a message in one logical step. Signcryption algorithms are deemed to be more efficient than an encryption and signature scheme because the former require less operations than the latter approach. ABSC provides fine-grained access control and data authenticity.

2.5 General framework of CP-ABE

A general framework of a ciphertext-policy attribute based encryption consists of four main algorithms: Setup, Encrypt, KeyGen, and Decrypt [20], that are described below.

- $\texttt{Setup}(\lambda, U) \rightarrow (MK, PK)$. The setup algorithm receives as input the security parameter λ (often in unary expression: 1^λ) and a universe of attributes U. The algorithm outputs the master key MK and the public parameters PK.
- $\texttt{Encrypt}(PK, m, \mathbb{A}) \rightarrow CT$. The inputs of the encryption algorithm are the public parameters PK, the message to be encrypted m, and the access structure or policy $\mathbb{A}$ that is defined over the attribute universe U. The output of the algorithm is the ciphertext CT such that only those users whose attributes satisfy the access structure $\mathbb{A}$ will be able to decrypt to obtain m. It is assumed that the access structure $\mathbb{A}$ is included implicitly in CT.
- $\texttt{KeyGen}(MK, S) \rightarrow SK$. The inputs of the key generation algorithm are the master key MK and an attribute set S. It outputs a private key SK such that the user in possession of SK is able to decrypt all the ciphertexts whose access structure $\mathbb{A}$ is satisfied by the attribute set S.
- $\texttt{Decrypt}(PK, CT, SK) \rightarrow m$. The decryption algorithm receives as inputs the public parameters PK, a ciphertext CT containing an access policy $\mathbb{A}$, and a private key SK that is associated to an attribute set S. If the attribute set satisfied the access policy, then the decryption algorithm returns the plaintext message m.

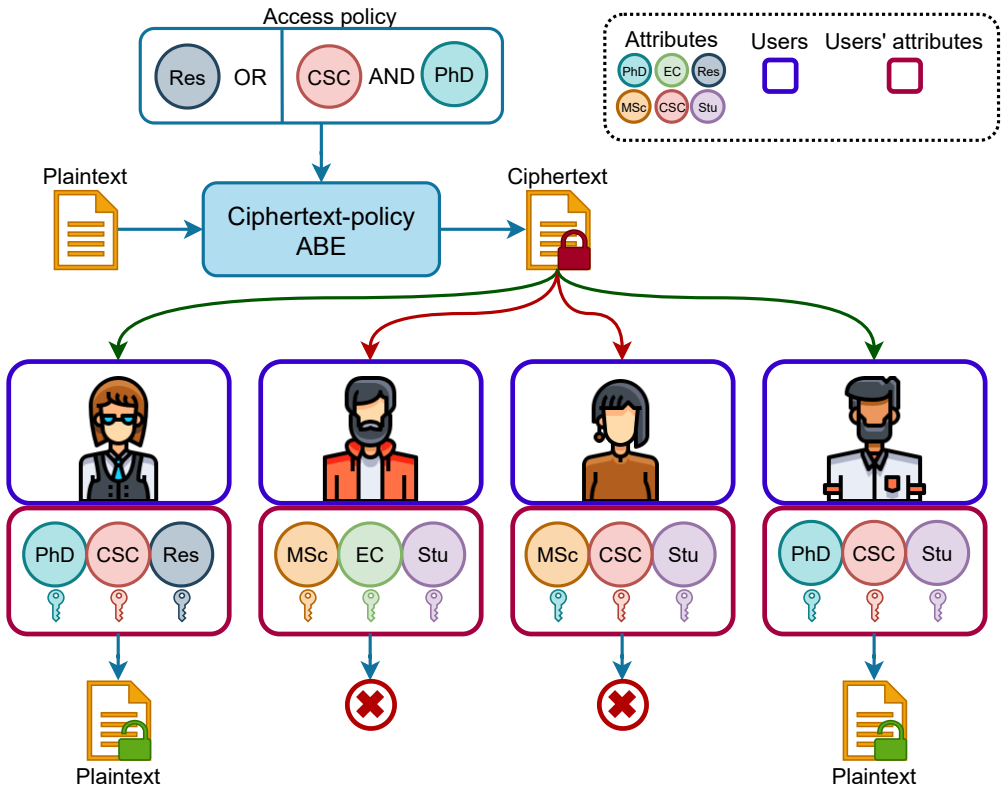

Figure 4. Example of a CP-ABE scheme. Only the users with secret keys associated to attributes that satisfy the access policy can decrypt the message.

2.6 Access structures

Definition 2.1 (**Access structure** [21])**.** Let $\{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}}$ is *monotone* if $\forall B, C$ if $B \in \mathbb{A}$ and $B \subseteq C$ implies that $C \in \mathbb{A}$. Then, an access structure (respectively, a monotone access structure) is a collection (respectively, a monotone collection) $\mathbb{A}$ of non-empty subsets of the set of parties $\{P_1, P_2, \ldots, P_n\}$, that is, $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}} \setminus \{\emptyset\}$. The sets that are in $\mathbb{A}$ are called the *authorized* sets, while the sets that are not in $\mathbb{A}$ are called the *unauthorized* sets.

2.6.1 Access trees

Definition 2.2 (**Access tree** [17, 22])**.** Let $\mathcal{T}$ be a tree representing an access structure or an access policy. Every non-leaf node of $\mathcal{T}$ represents a threshold gate according to a description of its children and a threshold value. Let x be a node of $\mathcal{T}$, num_x the number of children nodes of x, and k_x its threshold value, such that $0 < k_x \leq num_x$. A node x can represent an OR gate when $k_x = 1$, and an AND gate if we define $k_x = num_x$. Every leaf node of $\mathcal{T}$ is tagged with an attribute and k_x

is defined as $k_x = 1$.

As described in [22], some functions are commonly defined in order to operate appropriately the access tree. The function $\texttt{parent}(x)$ returns the parent of the node x. If a node x is a leaf node, the function $\texttt{att}(x)$ returns the attribute of the node x. It is useful to define an ordering between the children of every node in the tree numbering them from 1 to num_x. Thus, the number associated to the node x in this ordering is returned by the function $\texttt{index}(x)$.

Satisfying an access tree. Let r be the root node of an access tree $\mathcal{T}$. We can define the sub-tree of $\mathcal{T}$ rooted at the node x as $\mathcal{T}_x$. Thus, the access tree $\mathcal{T}$ can be also denoted as $\mathcal{T}_r$. Let S denote a set of attributes. We denote $\mathcal{T}_x(S) = 1$ if the set of attributes S satisfies the access tree $\mathcal{T}_x$. $\mathcal{T}_x(S)$ is executed recursively. If a node x is a non-leaf node, then $\mathcal{T}'_x(S)$ is evaluated for all children node x' of the node x. An 1 is returned by $\mathcal{T}_x(S)$ if and only if at least k_x children of x return 1. On the other hand, if x is a leaf node, then $\mathcal{T}_x(S)$ returns 1 if and only if the attribute attached to x belongs to the attribute set, that is, if $\texttt{att}(x) \in S$. Finally, a graphical example of an access structure represented as an access tree is depicted in Fig. 5.

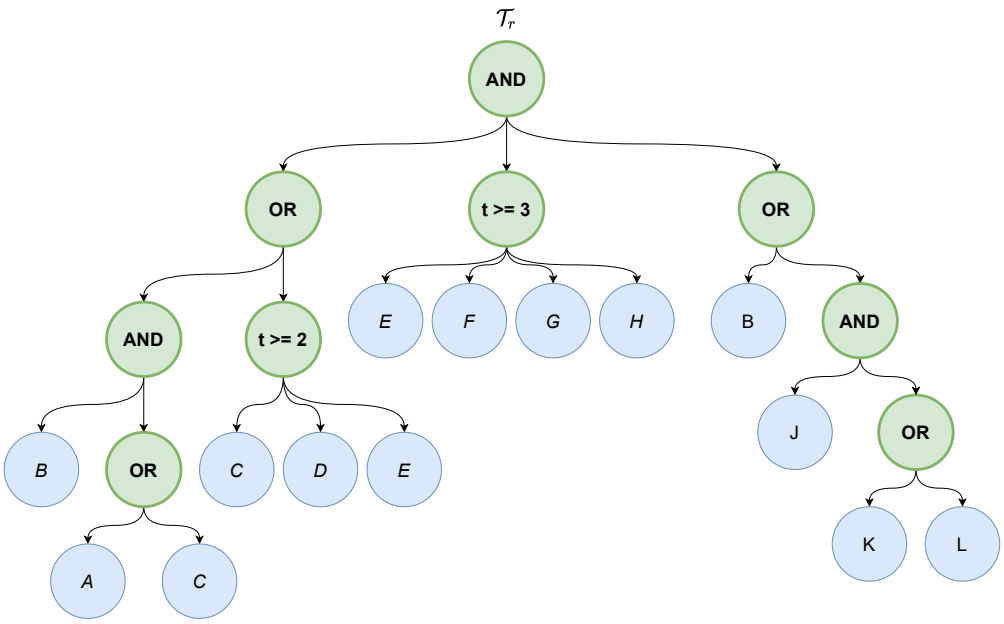

Figure 5. Example of an access tree rooted in the node $\mathcal{T}_r$ in a graphical form. The attribute universe is represented with the alphabet letters from A to L. The leaf nodes are the attributes of the access structure. The non-leaf nodes are threshold gates where t is the threshold value of the node. Nodes labeled as AND and OR, are also threshold gates with t equal to the number of child nodes for AND gates, and with t equal to 1 for OR gates.

2.6.2 Secret sharing scheme

Definition 2.3 (**Secret sharing scheme** [23])**.** Let S be a finite set of secrets. A distribution scheme Π with domain of secrets S realizing an access structure $\mathbb{A}$ is a secret-sharing scheme if the following requirements hold:

1. **Correctness.** The secret s can be reconstructed by any authorized set of parties. That is, for all set $B \in \mathbb{A}$ (with $B = \{P_{i_1}, \ldots, P_{i_{|B|}}\}$), there exists a function $\text{RECON}_B : S_{i_1} \times \ldots \times S_{i_{|B|}} \to S$, called reconstruction function, such that for every secret $s \in S$, and every random input r,
$$\Pr\left[\text{RECON}_B(\Pi(s,r)_B) = s\right] = 1.$$
2. **Perfect privacy.** In terms of information theory, every unauthorized set cannot learn anything about the secret. That is, for any set $B \notin \mathbb{A}$, for every two secrets $a, b \in S$, and for any possible vector of shares $\langle k_j \rangle_{P_j \in B}$:
$$\Pr\left[\Pi(a,r)_B = \langle k_j \rangle_{P_j \in B}\right] = \Pr\left[\Pi(b,r)_B = \langle k_j \rangle_{P_j \in B}\right]$$

2.6.3 Linear secret sharing schemes

Definition 2.4 (**Linear secret sharing scheme (LSSS)** [20,21])**.** Let Π be a secret sharing scheme over a set of parties $\mathcal{P}$. It is said that Π is a linear secret sharing scheme over $\mathbb{Z}_p$ if the following conditions hold:

1. The shares of each party $P \in \mathcal{P}$ form a vector over $\mathbb{Z}_p$.
2. There exists a $(l \times n)$ matrix A, called the share-generating matrix for Π, and there is a function $\rho(i) : \{1, \ldots, l\} \to \mathcal{P}$ that labels the i-th row of A with a party. Let $s \in \mathbb{Z}_p$ a secret to be shared using Π, and consider the column vector $v = (s, r_2, \ldots, r_l)$, where $r_2, \ldots, r_l \in \mathbb{Z}_p$ are random numbers. Then, the vector $\lambda = Av$ is the vector of l shares of the secret s following the scheme Π. The share λ_i belongs to the party indicated by $\rho(i)$.

LSSSs are used to represent access policies for ABE schemes. An example of an LSSS matrix M and a labeling function ρ is depicted in the following equation. The access policy represented by the access tree $\mathcal{T}_r$ of Fig. 5 has been transformed to an LSSS with a (14×8) matrix M using

the algorithm of Liu *et al.* [24].

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 3 & 0 & 0 & 0 \\ 1 & 2 & 4 & 0 & 0 & 1 & 1 & 0 \\ 1 & 2 & 4 & 0 & 0 & 2 & 4 & 0 \\ 1 & 2 & 4 & 0 & 0 & 3 & 9 & 0 \\ 1 & 2 & 4 & 0 & 0 & 4 & 10 & 0 \\ 1 & 3 & 9 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 9 & 0 & 0 & 0 & 0 & 1 \\ 1 & 3 & 9 & 0 & 0 & 0 & 0 & 2 \\ 1 & 3 & 9 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}, \ \rho(i) = \begin{bmatrix} B \\ A \\ C \\ C \\ D \\ E \\ E \\ F \\ G \\ H \\ I \\ J \\ K \\ L \end{bmatrix}$$

2.6.4 Bilinear maps

A bilinear maps are mathematical functions that are widely used to implement the so-called pairing-based cryptography. In this context, bilinear maps are also called bilinear pairings. The definition of these maps is given below.

Definition 2.5 (**Bilinear maps** [15, 22])**.** Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p. Let also g be a generator of the group $\mathbb{G}$. The map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is said to be bilinear if the following properties hold:

1. Bilinearity: for all $p, q \in \mathbb{G}$, and for all $a, b \in \mathbb{Z}_p$, we have $e(p^a, q^b) = e(p, q)^{ab}$.
2. Non-degeneracy: the map e does not map g to the identity $1_{\mathbb{G}_T}$ of $\mathbb{G}_T$, that is $e(g, g) \neq 1_{\mathbb{G}_T}$. On other words, if g is a generator of $\mathbb{G}$ then $e(g, g)$ is a generator of $\mathbb{G}_T$.
3. Computable: there exists an efficient algorithm to compute $e(p, q) \in \mathbb{G}_T$ for every $p, q \in \mathbb{G}$.

Groups $\mathbb{G}$ and $\mathbb{G}_T$ where such properties hold exist. The Tate and the Weil pairings are examples of these bilinear maps [25]. In these constructions, $\mathbb{G}$ is commonly an elliptic-curve group and $\mathbb{G}_T$ is a finite field.

2.7 Management mechanisms in ABAC

Attribute-based access control schemes may include some mechanisms to modify the state of the system in a given moment during its normal evolution. We refer as management mechanisms to those features of an ABAC scheme that allow to change the access rights of the system participants or modifying the access policies of the stored data. There are three of these management mechanisms that are mainly studied or proposed in the literature: user revocation, attribute update, and policy update, which are described as follows:

- **User revocation.** User revocation is the capacity of an access control scheme to remove the access rights of a given user of the system while the other users are still capable of accessing the stored data.
- **Attribute update.** The attribute update feature means that the scheme includes a mechanism to change the access rights of a given user by changing his/her attributes. This change in access privileges can be either by adding or reducing attributes of the user. Again, this process must not compromise the access capabilities of non-updated users in the system.
- **Access policy update.** The access policy update feature in ABAC schemes allows the data owner to change the access policies attached to an outsourced ciphertext. This change may include an addition or reduction of the attributes required to decrypt the data or even a change on the configuration of attributes in the access policy. This mechanism allows the scheme to respond to changes in the sensitivity of the data according to the preferences of the data owner.

2.8 Data aggregation

Data aggregation is the application of operations or transformations over a set of values to obtain an *aggregated* value with concentrated information about the original values [26]. An example of data aggregation is to compute one dimensional values from a vector of values such as the sum, product, mean, and variance. In the context of fog-enabled IoT applications data aggregation is an interesting mechanism to reduce memory usage for sensor nodes and minimize the communication overhead for the network.

2.8.1 Privacy-preserving data aggregation

One of the drawbacks of data aggregation mechanisms is the requirement of aggregator nodes that may compromise the privacy of data since they have access to different data sources or related to different users. For this reason, privacy-preserving data aggregation techniques have been proposed to overcome the concerns in this area [26–29]. The main approach of privacy-preserving data aggregation of homomorphic encryption that allows to perform arithmetic operations over encrypted data in such a way that the resulting ciphertext, when decrypted, is the same as the obtained when operations are performed in plaintext.

3 State of the art

The study of attribute-based access control schemes for fog-enabled IoT applications has received much attention in recent years. For this reason, this section presents a review of the most relevant works in ABAC schemes for the Fog-IoT environment. Those works based on attribute-based cryptography are mainly considered. The review process places special emphasis on systems that implement mechanisms for computation outsourcing, user and attribute revocation, and access policy update. Figure 6 shows a general diagram of the organization of the consulted literature.

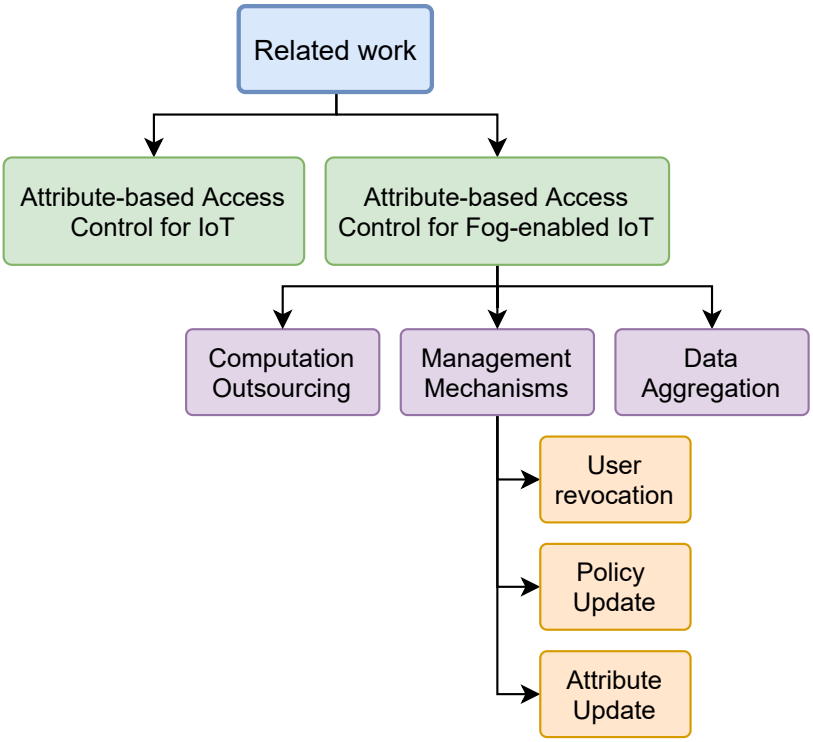

Figure 6. Overview of the organization of related work.

3.1 Attribute-based access control for IoT

The beginnings of ABAC for IoT can be traced back to the work of Yu *et al.* [30] in the context of a wireless sensor network (WSN). In their paper, a fine-grained access control scheme is proposed based on KP-ABE. In this system, the sensor nodes are assigned a set of attributes while users have in their possession an access structure. Sensed data is encrypted by sensor nodes using a symmetric key algorithm and the symmetric key is encrypted under the corresponding attributes. The scheme also incorporates collusion resistance and user revocation. However, the network model is highly constrained by the sensor nodes resources since they handle the user data access requests. Hu *et al.* [31] follow a similar approach and proposed a CP-ABE scheme to be used in wireless body area networks (WBANs) for patient-health monitoring. In this work, sensor nodes encrypt health reports using a symmetric encryption algorithm and encrypt the session key by CP-ABE. Keys and data are transmitted to a data sink where only those doctors, nurses, and authorized users with appropriate attributes can decrypt the reports. These proposals brought to light the benefits of attribute based access control mechanisms in the context of sensor networks.

Attribute-based encryption is a promising primitive for ABAC schemes that received much attention in recent years to protect data outsourced to the cloud [3]. This idea is exploited by Yao *et al.* [16], they proposed a lightweight KP-ABE access control scheme for data collected by IoT devices and outsourced to the cloud. The main idea is to achieve an efficient KP-ABE scheme suitable for resource constrained IoT devices by avoiding the expensive pairing operations used in previous

ABE schemes. Their scheme is based on elliptic curve cryptography (ECC) and is proven to be secure against chosen plaintext attack (CPA) under the Elliptic Curve Decisional Diffie-Hellman (ECDDH) assumption. Nevertheless, the scheme lacks the flexibility to revoke user attributes and to achieve scalability.

Eventually, more features were added to this kind of schemes. Yeh *et al.* [32] presented CP-ABE scheme for access control to personal health information. Data produced by lightweight IoT devices are outsourced to the cloud and protected with attribute based encryption to achieve fine-grained access control. This system incorporates auditing and attribute revocation mechanisms to increase the security and flexibility of the scheme. Following a similar path, the access control scheme proposed by Han *et al.* [33] focuses on a problem of user privacy. They pointed out that CP-ABE schemes may reveal sensitive information when the access policies, required to decrypt a message, are not hidden in a decryption query. Hence, they proposed a CP-ABE scheme for cloud-based IoT applications with policy hiding feature. Later, also for the cloud-IoT environment, Li *et al.* [34] presented an ABAC scheme going an step further including both hidden access polices and user accountability. The former is implemented to protect the user privacy, while the latter is used to avoid the key abuse problem in which a user is able share her/his secret keys to third parties.

Despite the relevant advances achieved in the area, a common problem in ABAC schemes in the cloud-IoT environment are the reduced computing resources available in the end user side. Thus, existing proposals of ABAC schemes made it clear that the development of more efficient access control schemes with new functionalities require different approaches, for instance, to leverage the fog computing paradigm. In the next section, a review of the most relevant ABAC schemes for the fog-enabled IoT environment is presented.

3.2 Attribute-based access control for fog-enabled IoT

Alrawais *et al.* [35] presented one of the first works involving the fog and ABE. They proposed a CP-ABE scheme to enable an efficient and secure key exchange protocol. Their idea is to use CP-ABE to exchange session keys for communications fog-to-cloud and fog-to-fog. The main drawback of the scheme is that IoT devices are not considered in the protocol. Moreover, no management mechanisms or network scalability are considered.

Taking into account the new threats in the fog environment, such as side channel attacks, Yu *et al.* [36] proposed a generic framework for leakage-resilient functional encryption (LR-FE) suitable for fine-grained access control schemes in fog computing. Their framework is based on functional encryption (FE) that is deemed to be a generalization of several cryptographic primitives, including IBE and ABE. Although, several instantiations of FE are obtained from this framework that are proven to be secure under the Continual Memory Leakage (CML) model, empirical performance evaluations are not reported. In addition, user revocation, policy update, and computation out-sourcing are not considered in the model which represents a significant limitation of the scheme in terms of flexibility and efficiency.

Eventually, research in this field led to novel security considerations. Jiang *et al.* [37] indicated that by using existing methods for defining access policies, a user can take subsets of its privileges to produce new secret keys and delegate them to unauthorized users or devices in the Fog-IoT environment, i.e., key delegation abuse. Attending this concern, the authors proposed a CP-ABE

scheme resistant to the key-delegation abuse problem and incorporates a traitor tracing mechanism. However, they achieve these features by sacrificing expressiveness and flexibility of the access policy. Additionally, the decryption process requires several pairing operations that are performed by the end devices which impacts the efficiency of the system.

3.3 Computation outsourcing

One of the main drawbacks of ABC schemes is that the involved operations, such as encryption or decryption, can be excessively expensive for resource-constrained IoT devices. The main reason is that the underlying cryptographic constructions are commonly based on pairing operations which may produce significant computational burden. An interesting approach to overcome this problem is to delegate costly operations to a more powerful node in the network, e.g., to a fog node. This idea has been previously used in the context of cloud-based data storage in [38] and [39]. Green *et al.* [38] introduced the concept of outsourced decryption for ABE using a proxy. On the other hand, Lounis *et al.* [39] only outsource the encryption process to a trusted gateway to reduce computation overhead in WSNs. Moreover, Lai *et al.* [40] introduced another important feature for ABE with outsourced decryption: verifiability of decrypted data. This feature refers to the verification of the integrity of received data when part or the total of the decryption process is delegated to another entity.

Given the clear advantages of the outsourcing approach for end devices and data owners, several access control schemes with cryptographic operations outsourced to the fog rapidly appeared. Zuo *et al.* [41] proposed an ABE scheme with decryption outsourcing and verification that is proven to be secure against chosen ciphertext attack (CCA). Similarly, in [42] a CP-ABE scheme with proxy decryption is proposed. In this work, the access policies are hidden to curious entities and multiple attribute authorities are considered to enable the scalability of the model and reduce the key escrow risk. Going one step further, Zhang *et al.* [43] presented an efficient CP-ABE scheme with outsourcing for both encryption and decryption operations achieving reduced and constant execution time for data owners and users. This approach is also applied for fog-enabled E-health applications by Zhao *et al.* [44]. In addition to outsourcing mechanisms for encryption and decryption, the scheme in [44] also includes the large universe property which means that the size of the common parameters of the scheme are no dependent to the size of the attribute universe. Furthermore, the initialization phase of this scheme does not require the specification of the total number of attributes to be used. This improves the flexibility and scalability of the ABAC scheme. Another CP-ABE scheme with outsourcing of the two operations is presented in [45]. Although the scheme does not include verification mechanisms, it integrates the attribute and user revocation features.

The strategy of outsourcing cryptographic operations also extended to schemes with multiple authorities. In the work of Vohra *et al.* [46], a CP-ABE scheme with multiple attribute authorities is proposed where each attribute authority is in charge of a subset of the attribute universe and issues the corresponding keys to the end users. This measure helps with the scalability of the system since the bottleneck of a unique attribute authority issuing the secret keys is avoided. In this work, only the decryption is outsourced to a proxy after a verification of the ciphertext, while the encryption is performed in two phases, namely offline and online encryption, to reduce the total

encryption time for the data owner.

Correspondingly, Belguith *et al.* [47] presented another CP-ABE scheme with multiple attribute authorities. In this case, the authors addressed the problem of publicly available access policies in ciphertexts and proposed a policy-hidden ABE scheme with decryption outsourcing. Access policies are composed of a generic attribute name and an attribute value. Generally, policies are transmitted along the ciphertext in the clear, leading to potential privacy leakages. Partially hidden policies were previously studied in [48, 49] as a measure to hide the attribute value. This approach conducted to CP-ABE with hidden access policies like the proposed in [50, 51].

Similarly to [47], Fan *et al.* [52] proposed a multi-authority (MA) CP-ABE scheme with hidden access policies and verifiable outsourced decryption. The system model considers two types of authorities: the certificate authority, and the attribute authority. The certificate authority is in charge of the registration of the attribute authorities and end users, and issues the corresponding identifiers for these entities. There are several independent attribute authorities in the system that have different attributes in their domain. They are in charge of hiding these attributes, issuing them to users, and generating the corresponding keys. Their results show that the efficiency on the final devices is improved compared to previous schemes. Finally, the main contribution of this work is the user revocation mechanism.

Outsourcing of both encryption and decryption with verification feature for CP-ABE with the setting of multiple authorities is achieved by Sun *et al.* in [53]. One of their main contributions is the hybridization of MA CP-ABE and searchable encryption (SE). This scheme achieves fine-grained access control by using expressive access policies based on a Linear Secret Sharing scheme (LSSS), and high computation efficiency for both data owners and users. Additionally, it is proven to be secure against the Chosen Plaintext Attack under the Decisional Bilinear Diffie-Hellman (DBDH) assumption. Similarly, Tu *et al.* [54] presented a MA CP-ABE scheme that allows to outsource encryption and decryption operations to the fog layer. The scheme also incorporates an attribute revocation mechanism based on attribute group keys. However, the verification of outsourced operations is not included in this proposal which leads to security concerns about the integrity of transmitted data.

Sarma *et al.* [55] also outsourced encryption and decryption of their CP-ABE scheme. In this case, only two main authorities are considered to avoid the key escrow problem: the key authority (KA) responsible for distributing public parameters of the scheme and issuing secret keys for data users; and the Attribute Manager (AM) who is in charge of joining users, key update, ciphertext re-encryption, and attribute addition or revocation. Although the scheme achieves efficiency for end devices and attributes revocation, it lacks a verification feature, and scalability is not considered.

Computation outsourcing is also present in access control schemes based in ABS and ABSC. Encryption and decryption operations are delegated to the fog nodes in the scheme of Huang *et al.* [56]. Their proposal is based in CP-ABE but additionally integrates an ABS mechanism that allows only authorized users to update the ciphertext after verifying their attributes. A similar situation can be found in the work of Zhang *et al.* [57] that proposed a CP-ABE scheme for the Internet of Vehicles (IoV). In this work, an ABS mechanism is applied to enable the verifiability of outsourced decryption. The scheme is efficient enough in the user side to be implemented in IoV applications.

In [58] a multi-authority access control scheme for the Fog-IoT environment based on attribute based signcrypion is presented. Signcryption and designcryption operations are delegated to the fog layer. Compared with previous ABSC schemes, the proposed one achieves high expressiveness in the access policies. In addition, the scheme allows to anonymously authenticate users in the network, and verification of the partially designcrypted data is also included.

On the other hand, a verifiable single authority CP-ABSC scheme for fine-grained access control is proposed in [59]. The main features of this scheme are the privacy-preserving data origin authentication, the access policy update, and the constant ciphertext size.

3.4 Management mechanisms

Management mechanisms for ABAC schemes in the Fog-IoT environment are an essential requirement due to the intrinsic dynamic nature of the involved nodes and entities. The network may experience a large number of joining and leaving nodes as a normal behavior. Furthermore, user privileges may vary over time according to their previous usage patterns or simply because they are upgraded or downgraded according to their active role. Similarly, there are situations where the access rules defined for the outsourced data must be changed. For instance, in the context of IIoT, data collected from a certain industrial process at a given point in time may not be as sensitive after some time has elapsed. In this situation, the data owner is entitled to change the data access policy in correspondence with this change in the sensitivity or importance of the data.

There are three main categories of management mechanisms considered in the literature: user revocation, attribute update (or revocation), and policy update [11, 12, 60]. Access control schemes that incorporate one or more of these features have been developed to respond to specific requirements of a given application environment. Some of the access control schemes based on attribute based cryptography that incorporate some type of management mechanism are described below.

3.4.1 User revocation

Some of the previously mentioned schemes offer the user revocation feature [42, 45, 52, 57]. In the work of Jahan *et al.* [42], two main groups of users are considered: writers and readers. The data owner is required to send two lists to the central authority with information about the readers and writers of data. Basically, there exists an access policy for reading data and another one for writing. When a user is revoked, the ciphertext is updated with modified attributes and corresponding secret values that are not issued to the revoked users. However, this revocation mechanism is not general for the overall system and each data owner is in charge of the revocation of readers and writers that have access to his/her data.

The scheme proposed by Chen *et al.* [45] achieves both user and attribute revocation. The system also outsources the encryption and decryption operations to fog nodes using partial proxy keys for each user. Thus, the revocation of a user only requires to delete the key for partial decryption from the fog nodes. This means that the end user will no longer be able to access the encrypted data because the fog node will no longer deliver its partially decrypted portion. The main advantages of this user revocation process, are that neither the group keys or the ciphertext need to be updated, reducing the computation overhead.

Only user revocation is implemented in the CP-ABE scheme of Fan *et al.* [52]. The system includes an anonymous attribute set that the CSP can link to users identifiers using a list in its possession. The algorithm for user revocation is executed by the CSP who, upon receiving a revocation request of a given user, removes the set of anonymous user attributes. The keys of the proxy server user for computation outsourcing are also deleted. Hence, the revoked user can no longer access data since the proxy cannot run the outsourced decryption algorithm. With this mechanism, it is not necessary to update the keys of non-revoked users or re-encrypting the ciphertexts.

Auditable user revocation is supported in the scheme of Zhang *et al.* [57] that is designed for applications of Internet of Vehicles (IoV). The system defines a group manager (GM) who is in charge of the revocation of users. This process includes a verification stage performed by an auditor (an entity) that decides if the data user has the required rights to request the revocation. Attribute based signatures are used to authenticate these revocation requests. Then, the GM writes the user identity into the revocation list and uses a version control to update group parameters. This method requires to re-encrypt the ciphertext and to update the private keys of remaining users in the group.

3.4.2 Policy update

The additional feature of updating the access policies allows increasing the flexibility of an access control scheme. The CP-ABSC scheme proposed by Belguith *et al.* [59] incorporates this feature by allowing the data owner to add or remove attributes to the access policies. Although the scheme is not intended for the fog computing paradigm, it includes a computation outsourcing mechanism to help resource-constrained IoT devices during the designcryption process with the verifiability property. The main advantage of this approach is that signcrypted data is combined with additional components and the policy updating only requires to modify these components. The CSP is in charge of the policy updating process which does not require to re-signcrypt the data or sending new secret keys to the system users. However, the system model in [59] relies on a single central trusted authority for issuing the private keys of the users and setting up the entire system, and this raises certain security and scalability concerns. Also, the underlying access structure is a threshold gate that presents a reduced ability to define precise fine-grained access policies, which impacts the expressiveness of the scheme.

Ling *et al.* [61] proposed another scheme with policy update. In this case, the authors are focused on designing a data access control scheme that supports: a large attribute domain, i.e., the size of public parameters does not depend on the number of attributes considered in the system; the policy update feature; white box traceability, i.e., the ability of tracking malicious users if they share their keys in an unauthorized way, and without requiring a list; multiple authorities; and high expressiveness. By including all these characteristics, the authors achieve a flexible and traceable access control scheme. However, the scheme does not include computation outsourcing mechanisms and leaves the end devices to perform the expensive encryption and decryption operations, which limits its applicability in resource-constrained devices.

3.4.3 Attribute update

As stated previously, attribute update refers the capacity of an attribute-based access control scheme to update the attributes of the users when their access rights change, whether they increase or decrease. In some works, only the second case is considered which is also known as attribute revocation. The scheme of Zhang *et al.* [43], includes the attribute update feature. In their approach, when an attribute of a given user is updated, not only the secret keys related to the attribute of the updated user is modified but also the keys of the non-updated users are changed. Also, the ciphertexts that include the updated attribute are updated so that they remain available in the system for future data requests.

The CP-ABSC scheme presented by Xu *et al.* [58] considers attribute revocation. There are multiple attribute authorities in the system, each of them is responsible of a different set of attributes. In the attribute revocation phase, the corresponding attribute authority chooses a new attribute version key and issues update keys to the non-revoked users and the CSP. Finally, the CSP is in charge of updating the ciphertexts tagged with the attributes that were updated.

In addition to user revocation, the scheme of Chen *et al.* [45] also includes attribute revocation. In this work, the key authority updates the group attribute that is delivered to the corresponding fog node. Then, the fog node produces the new attribute group key and makes a query to the CSP for updating the ciphertext. Then, future data requests are responded to by the cloud with the updated version of data.

Zhao *et al.* [44] studied the attribute revocation in their CP-ABE with verifiable computation outsourcing. They claim that, in their model, the attribute revocation only requires that the CSP reencrypts the ciphertext, replacing its access policy with another one, such that all the non-revoked users can decrypt the new ciphertext except the revoked user. However, the precise procedure to achieve this is not described which means that, if two or more users have the same set of attributes, the attribute revocation will potentially affect both.

Finally, Sarma *et al.* [55] presented a CP-ABE approach with attribute update. Specifically, they include mechanisms for attribute revocation and addition. The scheme includes system version and attribute version parameters that helps to define a history of the attributes and its validity along time. The attribute revocation phase affects the non-revoked users of the system by updating their keys that are related to the attribute that is revoked from a user. Again, this process requires to update the ciphertext tagged with the revoked attribute. On the other hand, in the attribute addition phase, the key authority issues the corresponding keys to the updated user who can only use her/his new attribute to decrypt ciphertexts produced after receiving the corresponding keys.

3.5 Data aggregation

The combination of attribute-based access control and privacy-preserving data aggregation has been studied in the work of Ruj *et al.* [62], whose full construction can be found in [63]. They proposed a security architecture to collect data from smart grids in a privacy preserving manner and allows data access control. The architecture is divided in two main parts. The first part consists of the smart meters in the home area network. Data produced by smart meters is collected and hierarchically aggregated using Paillier homomorphic encryption. The second part consists of

several remote terminal units that are in charge of sending the aggregated data to the CSP and defining the access policies to be applied to these data. Although this architecture offers one of the first constructions that join privacy-preserving data aggregation and an ABAC scheme, the two mechanisms work at different levels, one after the other.

In the context of cloud-assisted wireless body area networks for health applications, Fang *et al.* [64] proposed a CP-ABE scheme with data aggregation. In this work, the data collected from the sensors implanted in the patient body are blinded and then forwarded to a data sink that is in charge of the data aggregation process based on the Paillier cryptosystem. Then, the CP-ABE scheme is applied over aggregated data.

With regard to ABAC schemes with privacy-preserving data aggregation in fog-enabled IoT applications, to the best of the authors knowledge, the scheme proposed by Belguith *et al.* in [19] the only work that has included the data aggregation functionality to an ABAC scheme including the fog computing paradigm. It is based on CP-ABSC with a single attribute authority and no management mechanisms. In this scheme, IoT devices signcrypt the produced data that is later forwarded to an untrusted aggregator (fog node). The received data can be decrypted by the aggregator only if a minimum number of IoT devices cooperates. However, the scheme does not offer the possibility of defining access policies for individual data, and CSP entity is not considered for data storage outsourcing.

3.6 Discussion

A comparison of the most relevant reviewed schemes with their functionalities and characteristics is shown in Table 1. Although there are several features that can be used to describe these systems we focused on: the type of scheme, the type of access structure (AS), the use of multiple authorities (MA), if key escrow resistance (KER) is considered, supported management mechanisms, computation outsourcing mechanisms, for data aggregation (DA) support, the security model used to prove security, and if empirical performance evaluations (EPE) are reported.

There is a clear consensus in the literature on the benefits of CP-ABE to be used in ABAC schemes. The main reason behind is that this kind of cryptographic primitive allows to define expressive access policies and to clearly decide the properties (or attributes) that an authorized user should have. In addition, CP-ABE constructions are flexible enough to be integrated with additional functionalities that enhance their security, flexibility, and practicability. For these reasons, CP-ABE seems to be adequate for an access control scheme suitable for the fog-enabled IoT environment.

Four main approaches to define the underlying access structures of ABAC schemes are considered in the reviewed works: monotone access trees, linear secret sharing schemes, AND gates, and threshold gates. Among them, monotone access trees and LSSS are the most accepted and interesting approaches in the state of the art. Although both structures allow to define expressive access policies, the access tree representation is more meaningful for the end user. Alternatively, the LSSS approach provides a relatively compact representation that leads to efficient implementations while maintaining a good level of expressiveness.

Another consideration that is crucial for the behavior of an ABAC scheme for Fog-IoT applications is the number of authorities and the distribution of responsibilities. Several proposals use

Table 1. Comparison of existing access control schemes for the Fog-IoT environment. Acronyms: Access structure (AS), Multi-authority (MA), Key escrow resistance (KER), User revocation (UR), Policy update (PU), Attribute Update (AU), Encryption (Enc.), Decryption (Dec.), Verifiability (Ver.), Data aggregation (DA), Security model (SM), and Empirical performance evaluation (EPE).

Work	Type	AS	MA	KER	Management			Outsourcing			DA	SM	EPE
					UR	PU	AU	Enc.	Dec.	Ver.			
[35]	CP-ABE	Monotone Tree	×	×	×	×	×	×	×	×	×	–	✓
[36]	LR-FE	Monotone Tree	×	✓	×	×	×	×	×	×	×	CML	×
[37]	CP-ABE	AND gates	×	×	×	×	×	×	×	×	×	sCPA	×
[41]	CP-ABE	AND gates	×	×	×	×	×	×	✓	✓	×	CCA	✓
[42]	CP-ABE	Monotone Tree	✓	✓	✓	×	×	×	✓	×	×	CPA	✓
[43]	CP-ABE	Monotone Tree	×	×	×	×	✓	✓	✓	×	×	CPA	✓
[44]	CP-ABE	LSSS	×	×	×	×	✓	✓	✓	×	×	sCPA	✓
[45]	CP-ABE	Monotone Tree	×	×	✓	×	✓	✓	✓	×	×	–	✓
[46]	CP-ABE	Monotone Tree	✓	✓	×	×	×	×	✓	✓	×	–	✓
[47]	CP-ABE	LSSS	✓	✓	×	×	×	×	✓	✓	×	RCPA	✓
[52]	CP-ABE	LSSS	✓	✓	✓	×	×	×	✓	✓	×	–	✓
[53]	CP-ABE	LSSS	✓	✓	×	×	×	✓	✓	✓	×	CCA	✓
[54]	CP-ABE	LSSS	✓	✓	×	×	×	✓	✓	✓	×	CCA	✓
[55]	CP-ABE	Monotone Tree	×	✓	✓	×	✓	✓	✓	×	×	CPA	✓
[56]	CP-ABE / ABS	Monotone Tree	×	×	×	×	×	✓	✓	×	×	CPA	✓
[57]	CP-ABE / ABS	Monotone Tree	×	×	✓	×	×	×	✓	✓	×	CPA	✓
[58]	CP-ABSC	Span Program	✓	✓	×	×	✓	✓	✓	✓	×	CCA	✓
[59]	CP-ABSC	Threshold	×	×	×	✓	×	×	✓	✓	×	RCCA	✓
[61]	CP-ABE	LSSS	✓	✓	×	✓	×	×	×	×	×	sCPA	✓
[19]	CP-ABSC	Threshold	×	×	×	×	×	×	×	×	✓	–	✓

only a single authority in the system [19, 35–37, 41, 43–45, 56, 57, 59] which is mainly in charge of setting up the system and issuing the attributes and keys to the other entities and users. This approach is not suitable for highly dynamic environments since the central authority represents a single point of failure and it can be a bottleneck. In addition, there are security concerns in the single authority setting, for instance, the key escrow problem [55]. This problem refers to the capacity of the key authority to generate its own keys on behalf of users on the network, and then being able to decrypt all the stored ciphertexts. In this sense, the key escrow problem represents a threat to confidentiality and privacy of outsourced data. Therefore, it is essential to an access control scheme for the Fog-IoT setting to be key escrow resistant. The multi-authority approach is a promising measure to provide this feature. In addition, the distribution of responsibilities among different attribute or key authorities reduce the security risks and the dependence of the network

on a unique entity. Finally, the existence of multiple entities to manage the entry and exit of users is favorable to achieve the scalability of the access control scheme.

As mentioned earlier, IoT applications are commonly deployed in dynamic scenarios where the system may experience several changes during its operation. Hence, access control schemes in this environment require a certain level of flexibility to adapt to these changes. The incorporation of management mechanisms to access control schemes is a way of achieving this desired flexibility. The user revocation is the one of the main requirements for any access control scheme to revoke the access rights to malicious users. In addition, one of the situations that may arise is that, for different reasons, the privileges or the position (role) of a user in the system change. Then the scheme must be capable of reacting to this behavior and deliver the corresponding attributes and keys to that user, i.e., attribute update or, if the user only loses attributes, attribute revocation.

One of the main benefits of ABAC schemes based on attribute based cryptography is that the control over outsourced data is transferred to the data owner, who is responsible for defining access policies in a fine-grained mode. With a normal system operation, data produced by different types of devices may suffer a change on the sensitivity, i.e., the importance of these data may vary over time. In this sense, to achieve a flexible access control scheme, it is important that the user has the ability to modify the access policies defined for the ciphertexts, that is, policy update. Although this feature is included in some proposals of the literature [59, 61], its combination with other features and management mechanisms for ABAC schemes is still not well-studied.

In ABAC schemes based on ABC, the attribute update or revocation has not been studied as much as user revocation. For this reason, although for user revocation there exist efficient and practical solutions in the literature, most of the existing schemes with attribute revocation still require to update the ciphertexts and secret keys of the non-updated users that are related to the updated attributes. Such a process represents a drawback for ABAC schemes in terms of the scalability and processing efficiency of the system. Hence, more appropriate solutions are needed to obtain more flexible attribute-based access control schemes with wide set of management mechanisms.

We consider that one of the most important functionalities that the fog computing paradigm can offer to IoT applications is computation outsourcing. Particularly, the outsourcing of encryption and decryption procedures performed in an ABAC scheme that commonly include expensive operations. Even though these features have been widely investigated so far, it is critical to incorporate them in applications that involve resource-constrained IoT devices. Furthermore, in order to not compromise the security and reliability of a flexible fine-grained access control scheme, the verifiability of partially decrypted data must be also included.

Finally, in several IoT applications such as IIoT, the privacy-preserving data aggregation is an interesting and useful functionality that is desirable to include in access control schemes. There are two main reasons for this. First, the capability of aggregating collected data from several sensors can further increase the granularity of the access control over the stored data, allowing to certain users to only read aggregated data. Second, the memory and communication costs. Aggregated data requires much less space to be stored and, consequently, to be communicated. This last point is highly related to the scalability of the system.

4 Research proposal

4.1 Motivation

The emergence of the fog computing paradigm has brought a new ecosystem in which IoT applications can benefit in many ways. Existing solutions for security and privacy issues in the Cloud-IoT environment can be enhanced in this novel setting with either efficiency improvements or additional features [2]. For instance, security mechanisms to protect data generated by IoT devices and outsourced to the cloud can be further improved in terms of efficiency and availability for data owners and users.

Access control is an important measure to ensure the privacy of outsourced data to the cloud. Although a variety of access control schemes have been proposed for the cloud-based IoT environment there are still diverse challenges for this technology to meet several requirements in terms of security, flexibility, scalability, and other functionalities [2, 3, 10, 11].

Industrial Internet of Things (IIoT) is a subset of IoT focused on the interconnection of industrial elements and collection of data used with the aim of increasing production benefits, reducing energy consumption, monitoring assets and processes, among other goals [65]. Existing security and privacy issues for conventional IoT, can be more serious in IIoT, specially for the implications of data leakage or infrastructure damage.

Consequently, the study of methods and algorithms to develop secure and efficient fine-grained access control schemes suitable for the Fog-IoT environment is an important research area. Furthermore, the investigation of additional features for access control schemes, such as data aggregation, can further improve their suitability for promising applications such as smart homes/building/grids and the IIoT. For these reasons, in this project an attribute-based access control scheme for the fog-enabled IoT environment is proposed considering properties such as flexibility with management mechanisms, scalability, and data aggregation.

4.1.1 Example application scenario

In the context of IIoT, attribute-based access control provides a very useful way for industries to share the data they generate about their industrial processes with third parties or even within their own workforce. In this case, the industrial organization plays the role of the main data owner for which there are a large number of devices, or sensors, acting as data generators that produce the messages to be encrypted. The operators responsible for these devices are the data owners. In order to exploit the deployed sensor network, organizations may store their data in the cloud through data owners applying an attribute-based access control scheme. The fog computing paradigm is also exploited here for computation outsourcing, in particular, for data encryption and decryption. In this way, a Fog-IoT environment is established.

In order to obtain useful information both for decision making and for its use in facility operations, the organization requires data to be processed and analyzed by different subjects or entities that, depending on the needs of the company and the sensitivity of the data, may have different access rights and even access only to statistical information of the collected data. In this situation,

an ABAC deployed in this environment that includes appropriate management mechanisms and the privacy-preserving data aggregation feature would handle such requirements.

In the mentioned setting, end users are internal or external data analysts that clearly may have different access rights depending on the requirements of the data process. Actuator devices may also need access to a specific type of data to properly perform their functions. Hence, the attribute universe may contain fields such as association (internal or external), location, facility, type of process, type of analyst, type of receiver, among others. A model of the example application scenario is presented in Fig. 7.

Finally, the growing nature of this environment raises the importance of considering a scalable system model in which multiple attribute authorities can manage the generation of keys and modification of access rights. Additionally, in order to establish a certain level of flexibility it is worthwhile to include mechanisms for user revocation, attribute and access policy update, and the possibility of updating the access policies of encrypted data.

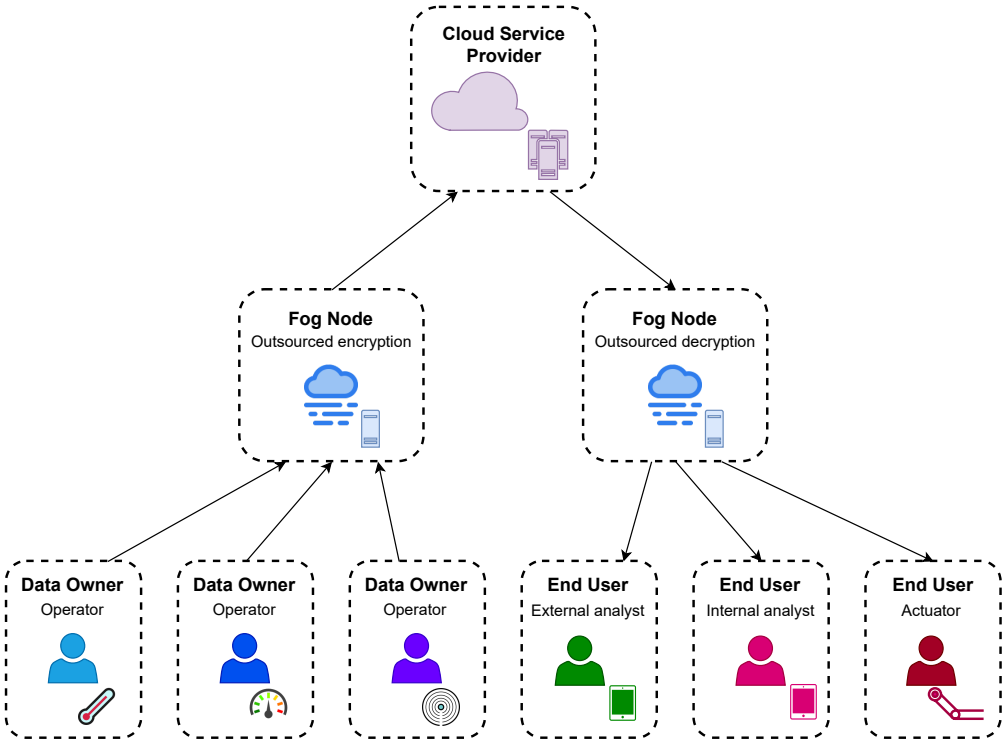

Figure 7. A model of the example application scenario: fine-grained access control scheme for IIoT.

4.2 Justification

In the context of access control schemes in the Fog-IoT environment, the following are detected issues that will be addressed in this project:

1. **Management mechanisms:** The dynamic nature of the mentioned environment requires measures to ensure security and efficiency that consider the joining and leaving of end devices, fog nodes, and users to the network. Thus, this problem includes verifying the validity of users' attributes, and that their privileges are up to date. In addition, this problem includes the user revocation problem and the update of the access policies, i.e., the access structures than an authorized user must fulfill in order to decrypt a given ciphertext.

2. **Scalability:** This problem refers to the ability of an access control scheme to maintain security and efficiency as the number of joining entities grows.

3. **Secure and privacy-preserving data aggregation:** This problem refers to the adaptation of a data aggregation mechanism to an access control scheme which preserves data privacy and the security of the scheme.

4.3 Problem statement

In order to obtain an attribute-based access control scheme suitable for the fog-enabled IoT environment, the following challenges must be addressed:

1. Considering that the trust in users and entities may change over time, an access control scheme must provide a management mechanism that is capable of revoking users, updating attributes of users, and updating access policies while maintaining data privacy and a reduced impact on the efficiency of the overall scheme.

2. The scheme must consider the inherent growing nature of the Fog-IoT environment and support the scalability while maintaining the security and efficiency of the system.

3. To reduce communication overhead and further preserve the privacy over outsourced data, the access control scheme must provide a privacy-preserving data aggregation mechanism in which the data owner has control over the aggregated data.

Formally: Let Π be an attribute based access control scheme, a_k be the k-th attribute of the attribute universe $\mathcal{U}$, n be the number of users, $U_{i\in\{1,\ldots,n\}}$ denote the i-th user, S_{U_i} be the attribute set assigned the user U_i and SK_{U_i} the associated secret key, and CT be a ciphertext that implicitly contains an access policy $\mathbb{A}$. Thus, this research project addresses the problem of achieving a data access control scheme Π that includes algorithms: to revoke any user U_i in the system, i.e., U_i cannot longer use SK_{U_i} to access any CT stored in a CSP and encrypted by Π; to update the attribute set S_{U_i} of the user U_i, i.e., if a_k is the attribute to be added/revoked to/from S_{U_i}, then SK_{U_i} is updated to a new SK'_{U_i} that reflects the addition/deletion of a_k; to change any $\mathbb{A}$ attached to a given CT to a new $\mathbb{A}'$ that reflects a different access policy; and to allow the privacy-preserving data aggregation, i.e., if $CT_1, \ldots, CT_d$ are d ciphertexts of the same kind, then they can be aggregated into a ciphertext $\overline{CT}$ that, when decrypted reveals only statistical information about the original ciphertexts. The aforementioned while maintaining the security and privacy of encrypted data, and responding efficiently to the increase in the number of users.

4.4 Research questions

The following questions will guide the development of this research:

1. What features should a management mechanism for an attribute-based access control scheme have so that user revocation, attribute update, and policy update operations have a reduced impact on the efficiency of the system?

2. How can an attribute-based access control scheme be defined in the Fog-IoT environment so that it is scalable, flexible, and secure?

3. How can attribute-based data access control and privacy-preserving data aggregation be integrated into a cryptographic algorithm-based scheme?

4.5 Hypothesis

Given an access control scheme for the Fog-IoT environment based on attribute-based cryptography it is possible to construct a wider scheme that incorporates a management mechanism for attribute update, policy update, and user revocation which also supports the privacy-preserving data aggregation feature.

4.6 Main objective

To develop an attribute-based access control scheme for data produced by devices in the Fog-IoT environment that includes a management mechanism for attribute update, access policy update, and user revocation, supports privacy-preserving data aggregation, and is scalable in terms of supporting an increasing number of users in the system.

4.6.1 Specific objectives

1. To develop a secure and scalable attribute-based data access control scheme for Fog-enabled IoT applications.

2. To design a management mechanism for updating attributes, access policies, and revoking users.

3. To design a privacy-preserving data aggregation algorithm that is compatible with the access control scheme.

4. To integrate the developed algorithms and mechanisms to obtain a scalable and flexible access control scheme with data aggregation capability.

4.7 Scope and limitations

- The research is concerned with cryptographic algorithms for access control and data aggregation for data produced by IoT devices.
- It is assumed that encryption and decryption algorithms are applied to data or messages with a relatively short-length representation, not for direct application on files.
- The notion of efficiency used in this work is related to the execution time in fog nodes and resource constrained IoT devices.
- It is assumed that there exist secure channels for communication of entities.
- The key-agreement protocol, authentication of users or entities, and the criteria for attribute assignment are out of the scope of this work.

4.8 Expected contributions

- A management mechanism for attribute-based access control schemes.
- A data aggregation algorithm compatible with attribute-based access control schemes.
- A scalable attribute-based access control scheme suitable for fog-enabled IoT applications.

4.9 Methodology

In order to accomplish the mentioned objectives, the research project will be directed by the following five stages.

4.9.1 Analysis of existing access control schemes for fog-enabled IoT applications

In this stage, the existing approaches of access control schemes will be studied. The main goal is to identify and analyze in the different schemes:

- The cryptographic constructions and primitives used to enable the proposed scheme.
- The security services provided by each particular scheme, as well as the security model (if applicable) and assumptions used to define the security level.
- Mechanisms (if implemented) for computation outsourcing and system management mechanisms including user revocation, attribute revocation, and policy update.
- Performance of the overall scheme and the effects on each involved entity (in terms of computation efficiency).

4.9.2 Development of attribute-based access control scheme

The problems and drawbacks of existing approaches found in the previous stage will be addressed in this stage. The main goal is to propose an access control scheme that fulfills the following requirements:

- **Security.** The scheme must securely provide different security services such as data confidentiality, authenticated access control, key escrow resilience, among others.
- **Scalability.** The proposed scheme must be scalable in terms of the number of entities and users considered in the scheme and the effects they produce for the computation overhead and the size of the required elements (ciphertexts, access policies, and keys). Considering that the Fog-IoT is a highly dynamic environment, the scheme must be able to respond adequately and efficiently to the growth of the network.
- **Expressivity.** This property is related to the level of detail that can be used in defining an access policy. The goal is to base the scheme in an access structure that enables a high level of detail or granularity in order to achieve the so-called fine-grained access control.

4.9.3 Development of management mechanism

The access control scheme developed in the previous stage must consider appropriate cryptographic primitives and algorithms that allow to include a management mechanism to ensure the flexibility of the scheme.

- **User revocation.** A user revocation algorithm that allows to limit the access to outsourced data to users specified by revocation requests is developed in this phase. The scheme Π must be able to prevent a revoked user U_i from using her/his key SK_{U_i} to decrypt any ciphertext CT even if U_i colludes with a malicious non-revoked user U_j in the system.
- **Attribute update.** The management mechanism must include the attribute update feature that allows to change the attribute set S_{U_i} of a given user U_i either by removing an attribute $a_k \in S_{U_i}$ or by adding an an attribute a_l to S_{U_i}. Hence, the key SKU_i must be updated in such a way it reflects the changes applied to S_{U_i}, while reducing as much as possible the effects on the rest of the users and outsourced ciphertexts.
- **Policy update.** The final step in this stage is the design of the policy update procedure of the management mechanism. Thus, the scheme must include an algorithm to process requests from a data owner to modify the access policy $\mathbb{A}$ attached to a given ciphertext CT already outsourced to the CSP. This operation must be executed in such a way the privacy of altered CT is preserved.

4.9.4 Development of privacy-preserving data aggregation algorithm

This stage is dedicated to the design of an additional capability for the access control scheme: data aggregation. The data aggregation algorithm must be compatible with the cryptographic constructions implemented in the developed access control scheme, and further preserve the privacy of the aggregated data.

5 Preliminary results

In this section, a brief description of the progress achieved so far is presented. In addition, preliminary results of the implementation of a reference scheme are reported.

5.1 Collection and study of development tools

Given that the current project is focused on access control schemes applicable to devices in the Fog-IoT environment, the cryptographic techniques and development tools must support the implementation on resource-constrained devices. For this reason, we have collected some implementation tools, used recurrently in the consulted literature, for the evaluation of the proposed algorithms.

5.1.1 MIRACL

MIRACL[1] is an open source software development kit for the C programming language that includes several cryptographic routines and a set of functionalities that can be used to construct number-theoretic-based algorithms. Additionally, the library has support for arbitrary precision arithmetic and allows to use different cryptographic primitives not only in a computer but is also designed for its use in embedded systems.

5.1.2 Pairing-based cryptography (PBC) library

The PBC library[2] is currently an standard for researchers that use pairings for cryptography. The library includes a wide range of functions and data structures to enable cryptographic schemes based on pairings or bilinear pairings. This is important because most of the state-of-the-art attribute-based access control schemes rely on bilinear pairings as the underlying cryptographic technique.

5.2 Implementation of a baseline CP-ABE scheme

In order to become familiar with the mentioned libraries a the baseline CP-ABE scheme, the scheme of Waters [20] was implemented. Our particular interest on this implementation is because it firstly expressed the access structures by a linear secret sharing scheme matrix which is an efficient representation of the access policies. In addition, there exist algorithms in the literature to efficiently transform an access tree with threshold gates into a LSSS, such as the algorithm of Liu *et al.* [24], which is desirable because an access structure represented by an access tree is more expressive than an LSSS matrix.

Objective. The main objective of this experiment is to obtain information about the effects on the execution time of different number of attributes in the access structure. The encryption and decryption times are compared against the number of attributes.

[1] `https://github.com/miracl/MIRACL`

[2] https://crypto.stanford.edu/pbc/

Method. The four algorithms Setup, Enc, KeyGen, Dec of the basic scheme of Waters presented in [20] were implemented using the MIRACL library. In addition, an implementation of the algorithm in [24] was used to transform access structures described as access trees into LSSS for efficiency. The programs were executed on a laptop computer equipped with Intel Core i5-8300 2.30 GHz, 16 GB of memory, and Ubuntu 20.04 operative system. In the following, the procedures Setup, Enc, KeyGen, and Dec of the scheme of Waters are shown in the Algorithms 1, 2, 3, and 4, respectively.

Algorithm 1 Setup (Setup)

Input: U: the attribute universe
λ: the security parameter
Output: PK: the public parameters
MK: the master secret key

procedure SETUP(U, λ)
  $\mathbb{G} \leftarrow$ a group of prime order p
  $\mathbb{G}_T \leftarrow$ a group of prime order p
  $e \leftarrow$ a bilinear map $e : \mathbb{G} \times \mathbb{G} \mapsto \mathbb{G}_T$
  $g \leftarrow$ a generator of $\mathbb{G}$
  $\{h_1, \ldots, h_U\} \leftarrow U$ elements of $\mathbb{G}$ randomly chosen
  $(\alpha, a) \leftarrow$ two random elements of $\mathbb{Z}_p$
  $PK \leftarrow (g, e(g,g)^\alpha, g^a, h_1, \ldots, h_U)$
  $MK \leftarrow g^\alpha$
  return PK, MK
end procedure

Access trees were defined as AND gates to force the scheme to require the maximum number of attributes for an authorized user and, consequently, to use the maximum number of array operations per number of attributes, i.e., we tested the worst scenario according to the number of used attributes. The number of attributes in the access policies ranges from 10 to 50 in increments of 5.

Parameters. The elliptic curve used for pairings is in the form of standard Weierstrass equation $y^2 = x^3 + ax + b \pmod p$, where p is a prime number congruent to $3 \pmod 4$ and the constant $a = -3$. The selected security parameter λ determines the bit length of p, and parameter q is the number of points in the curve which is also prime. Then, parameters of the curve are:

$$\lambda = 192$$
$$b = 4265732895672588129268258440977714335632089762934383523494$$
$$p = 4930024174431634640599033341057067222865862716297522433299$$
$$q = 4930024174431634640599033341125441632693811654341940586403$$

Results. The obtained results of encryption and decryption times are shown in Fig. 8. The execution times are in milliseconds and were obtained by averaging 100 executions. From these results it

Algorithm 2 Encrypt (Enc)

Input: PK: the public parameters
(M, ρ) the access policy as an $(l \times n)$ LSSS matrix and a labeling function
m: the message
Output: CT: the ciphertext

procedure ENC($PK, (M, \rho), m$)
  $v \leftarrow$ a vector $(s, v_2, \ldots, v_n) \in \mathbb{Z}_p^n$ where s is the encryption exponent
  for $i \leftarrow 1$ **to** l **do**
    $\lambda_i \leftarrow v \cdot M_i$ where M_i is i-th row vector of M
  end for
  $\{r_1, \ldots, r_l\} \leftarrow l$ random elements of $\mathbb{Z}_p$
  $C \leftarrow m \cdot e(g, g)^{\alpha s}$
  $C' \leftarrow g^s$
  for $i \leftarrow 1$ **to** l **do**
    $C_i \leftarrow g^{a\lambda_i} h_{\rho(i)}^{-r_i}$
    $D_i \leftarrow g^{r_i}$
  end for
  $CT \leftarrow (C_i, D_i)_{i \in 1, \ldots, l}$
  return CT and a description of (M, ρ)
end procedure

Algorithm 3 Key generation (KeyGen)

Input: MK: the master secret key
S: an attribute set
Output: SK: a private key

procedure KEYGEN(MK, S)
  $t \leftarrow$ a random element of $\mathbb{Z}_p$
  $K \leftarrow g^{\alpha} g^{at}$
  $L \leftarrow g^t$
  for all $a \in S$ **do**
    $K_a \leftarrow h_a^t$
  end for
  $SK \leftarrow (K, L, \{K_a\}_{\forall a \in S})$
  return SK
end procedure

Algorithm 4 Decrypt (Dec)

Input: CT: a ciphertext
SK: a private key for the attribute set S satisfying the access policy in CT
Output: m: the decrypted message

1: **procedure** DEC(CT, SK)
2: $I \leftarrow \{i : \rho(i) \in S\}$, i.e., the set of row indexes of M labeled by ρ with attributes of S
3: Compute the set of constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that if the set of shares $\{\lambda_i\}$ is valid for reconstructing the secret s according to M, then $\sum_{i \in I} \omega_i \lambda_i = s$
4: Compute $Rec = \frac{e(C',K)}{\prod_{i \in I}\left(e(C_i,L)e(D_i,K_{\rho(i)})\right)} = \frac{e(g,g)^{\alpha s}e(g,g)^{ast}}{\prod_{i \in I}\left(e(g,g)^{ta\lambda_i\omega_i}\right)} = e(g,g)^{\alpha s}$
5: $m \leftarrow C/Rec$, that is, $m = me(g,g)^{\alpha s}/e(g,g)^{\alpha s}$
6: **return** m
7: **end procedure**

is observed that decryption time is considerably higher than encryption and that the former grows faster than the latter with the number of attributes. However, from Fig. 8 we can infer that both operations grow linearly with the number of attributes which is one of the most interesting results of CP-ABE based on LSSS.

Conclusions The behavior of the implemented scheme indicates that attribute based encryption, despite its interesting ability to enable one-to-many cryptography, produces a significant computational burden for both encryption and decryption terminals. In our experiment, a personal computer with relatively common specifications is used to execute the two algorithms in a baseline CP-ABE scheme. Although the obtained execution times are small, for several applications in which the time is a decisive factor these results are not quite acceptable. Furthermore, this situation may be further aggravated when resource-constrained devices, such as IoT nodes, are used. In addition, given that the data is decrypted by a high number of users in an attribute-based access control scheme, and considering that the decryption process is more expensive than encryption, several problems may arise in such a scheme. For these reasons, it seems clear that the integration of fog-based computation outsourcing mechanisms, for access control schemes based on ABE for data produced by IoT devices, is required and must be considered in the proposed construction. In addition, for the sake of security, the data verification technique is also an interesting approach to protect the outsourced data.

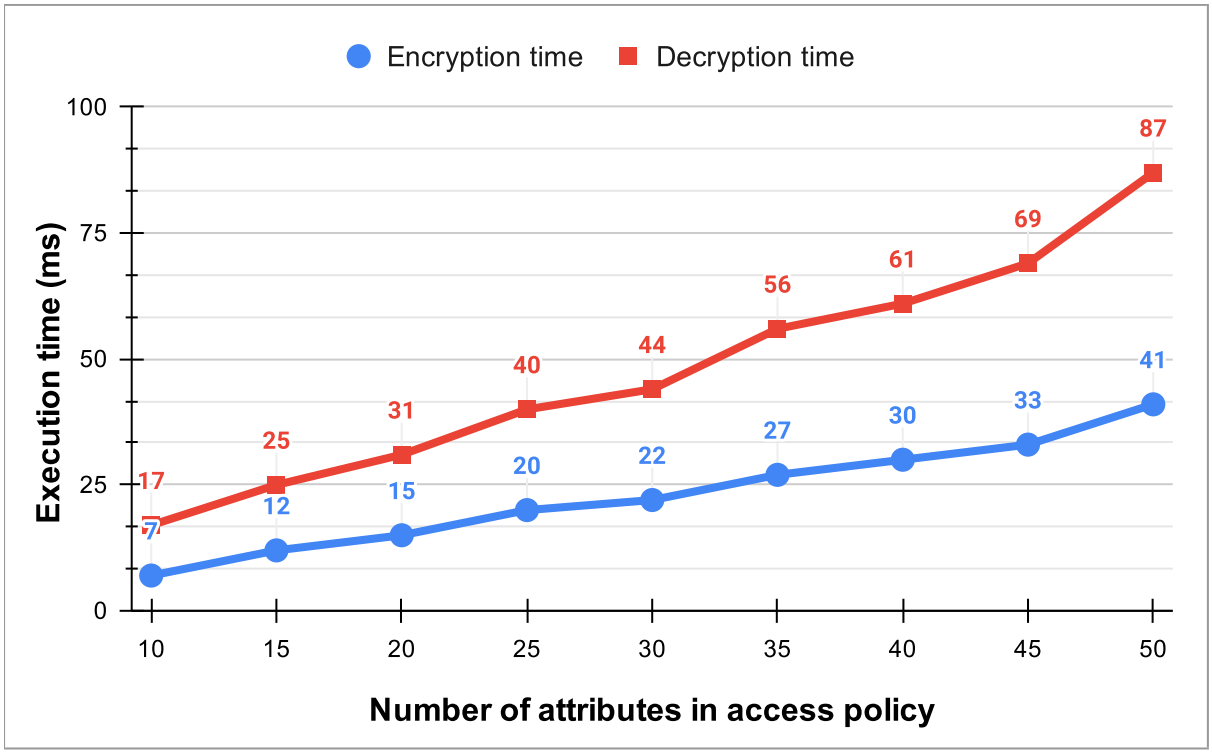

Figure 8. Encryption and decryption times for our implementation of the scheme of Waters [20].

References

[1] H. F. Atlam, R. J. Walters, and G. B. Wills, "Fog computing and the internet of things: A review," *big data and cognitive computing*, vol. 2, no. 2, p. 10, 2018.

[2] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Securing fog computing for internet of things applications: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2017.

[3] S. Belguith, N. Kaaniche, and M. Hammoudeh, "Analysis of attribute-based cryptographic techniques and their application to protect cloud services," *Transactions on Emerging Telecommunications Technologies*, p. e3667, 2019.

[4] C. MacGillivray and A. Wright, "Worldwide internet of things connectivity forecast, 2017–2021," 2017.

[5] 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.

[6] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, 2012.

[7] R. Z. Naeem, S. Bashir, M. F. Amjad, H. Abbas, and H. Afzal, "Fog computing in internet of things: Practical applications and future directions," *Peer-to-Peer Networking and Applications*, vol. 12, no. 5, pp. 1236–1262, 2019.

[8] 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*, vol. 6, no. 1, pp. 46–59, 2015.

[9] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.

[10] 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 Communications Magazine*, vol. 56, no. 2, pp. 144–149, 2018.

[11] M. A. Aleisa, A. Abuhussein, and F. T. Sheldon, "Access control in fog computing: Challenges and research agenda," *IEEE Access*, vol. 8, pp. 83986–83999, 2020.

[12] R. R. Al-Dahhan, Q. Shi, G. M. Lee, and K. Kifayat, "Survey on revocation in ciphertext-policy attribute-based encryption," *Sensors*, vol. 19, no. 7, p. 1695, 2019.

[13] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Springer, 2005.

[14] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Workshop on the theory and application of cryptographic techniques*, pp. 47–53, Springer, 1984.

[15] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Annual international cryptology conference*, pp. 213–229, Springer, 2001.

[16] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the internet of things," *Future Generation Computer Systems*, vol. 49, pp. 104–112, 2015.

[17] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98, 2006.

[18] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Cryptographers' track at the RSA conference*, pp. 376–392, Springer, 2011.

[19] S. Belguith, N. Kaaniche, M. Mohamed, and G. Russello, "Coop-daab: Cooperative attribute based data aggregation for internet of things applications," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pp. 498–515, Springer, 2018.

[20] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *International Workshop on Public Key Cryptography*, pp. 53–70, Springer, 2011.

[21] A. Beimel *et al.*, "Secure schemes for secret sharing and key distribution," 1996.

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

[23] A. Beimel, "Secret-sharing schemes: A survey," in *International conference on coding and cryptology*, pp. 11–46, Springer, 2011.

[24] Z. Liu, Z. Cao, and D. S. Wong, "Efficient generation of linear secret sharing scheme matrices from threshold access trees," *Cryptology ePrint Archive: Listing*, 2010.

[25] R. Rivest, "Lecture 25: Pairing-based cryptography," 2004.

[26] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced iot," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.

[27] R. Bista, K.-J. Jo, and J.-W. Chang, "A new approach to secure aggregation of private data in wireless sensor networks," in *2009 eighth IEEE international conference on dependable, autonomic and secure computing*, pp. 394–399, IEEE, 2009.

[28] S. B. Othman, A. A. Bahattab, A. Trad, and H. Youssef, "Confidentiality and integrity for data aggregation in wsn using homomorphic encryption," *Wireless Personal Communications*, vol. 80, no. 2, pp. 867–889, 2015.

[29] X. Li, S. Liu, F. Wu, S. Kumari, and J. J. Rodrigues, "Privacy preserving data aggregation scheme for mobile edge computing assisted iot applications," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4755–4763, 2018.

[30] S. Yu, K. Ren, and W. Lou, "Fdac: Toward fine-grained distributed data access control in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 4, pp. 673–686, 2010.

[31] C. Hu, H. Li, Y. Huo, T. Xiang, and X. Liao, "Secure and efficient data communication protocol for wireless body area networks," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 2, pp. 94–107, 2016.

[32] L.-Y. Yeh, P.-Y. Chiang, Y.-L. Tsai, and J.-L. Huang, "Cloud-based fine-grained health information access control framework for lightweight iot devices with dynamic auditing and attribute revocation," *IEEE transactions on cloud computing*, vol. 6, no. 2, pp. 532–544, 2015.

[33] Q. Han, Y. Zhang, and H. Li, "Efficient and robust attribute-based encryption supporting access policy hiding in internet of things," *Future Generation Computer Systems*, vol. 83, pp. 269–277, 2018.

[34] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, and D. Wang, "Attribute based encryption with privacy protection and accountability for cloudiot," *IEEE Transactions on Cloud Computing*, 2020.

[35] 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.

[36] Z. Yu, M. H. Au, Q. Xu, R. Yang, and J. Han, "Towards leakage-resilient fine-grained access control in fog computing," *Future Generation Computer Systems*, vol. 78, pp. 763–777, 2018.

[37] Y. Jiang, W. Susilo, Y. Mu, and F. Guo, "Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing," *Future Generation Computer Systems*, vol. 78, pp. 720–729, 2018.

[38] M. Green, S. Hohenberger, B. Waters, *et al.*, "Outsourcing the decryption of abe ciphertexts.," in *USENIX security symposium*, vol. 2011, 2011.

[39] A. Lounis, A. Hadjidj, A. Bouabdallah, and Y. Challal, "Healing on the cloud: Secure cloud architecture for medical wireless sensor networks," *Future Generation Computer Systems*, vol. 55, pp. 266–277, 2016.

[40] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on information forensics and security*, vol. 8, no. 8, pp. 1343–1354, 2013.

[41] C. Zuo, J. Shao, G. Wei, M. Xie, and M. Ji, "Cca-secure abe with outsourced decryption for fog computing," *Future Generation Computer Systems*, vol. 78, pp. 730–738, 2018.

[42] M. Jahan, S. Seneviratne, B. Chu, A. Seneviratne, and S. Jha, "Privacy preserving data access scheme for iot devices," in *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, pp. 1–10, IEEE, 2017.

[43] 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 Generation Computer Systems*, vol. 78, pp. 753–762, 2018.

[44] J. Zhao, P. Zeng, and K.-K. R. Choo, "An efficient access control scheme with outsourcing and attribute revocation for fog-enabled e-health," *IEEE Access*, vol. 9, pp. 13789–13799, 2021.

[45] S. Chen, M. Wen, R. Lu, J. Li, and S. Chen, "Achieve revocable access control for fog-based smart grid system," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pp. 1–7, IEEE, 2019.

[46] K. Vohra and M. Dave, "Multi-authority attribute based data access control in fog computing," *Procedia computer science*, vol. 132, pp. 1449–1457, 2018.

[47] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, "Phoabe: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot," *Computer Networks*, vol. 133, pp. 141–156, 2018.

[48] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *International conference on applied cryptography and network security*, pp. 111–129, Springer, 2008.

[49] J. Lai, R. H. Deng, and Y. Li, "Expressive cp-abe with partially hidden access structures," in *Proceedings of the 7th ACM symposium on information, computer and communications security*, pp. 18–19, 2012.

[50] T. V. X. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute-based encryption under standard assumptions," *IEEE transactions on information forensics and security*, vol. 11, no. 1, pp. 35–45, 2015.

[51] R. Xu and B. Lang, "A cp-abe scheme with hidden policy and its application in cloud computing," *International Journal of Cloud Computing*, vol. 4, no. 4, pp. 279–298, 2015.

[52] K. Fan, H. Xu, L. Gao, H. Li, and Y. Yang, "Efficient and privacy preserving access control scheme for fog-enabled iot," *Future Generation Computer Systems*, vol. 99, pp. 134–142, 2019.

[53] 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, p. e0207543, 2018.

[54] S. Tu, M. Waqas, F. Huang, G. Abbas, and Z. H. Abbas, "A revocable and outsourced multi-authority attribute-based encryption scheme in fog computing," *Computer Networks*, vol. 195, p. 108196, 2021.

[55] R. Sarma, C. Kumar, and F. A. Barbhuiya, "Pac-fit: An efficient privacy preserving access control scheme for fog-enabled iot," *Sustainable Computing: Informatics and Systems*, p. 100527, 2021.

[56] 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.

[57] J. Zhang, T. Li, M. S. Obaidat, C. Lin, and J. Ma, "Enabling efficient data sharing with auditable user revocation for iov systems," *IEEE Systems Journal*, 2021.

[58] 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, 2018.

[59] S. Belguith, N. Kaaniche, M. Hammoudeh, and T. Dargahi, "Proud: Verifiable privacy-preserving outsourced attribute based signcryption supporting access policy update for cloud assisted iot applications," *Future Generation Computer Systems*, vol. 111, pp. 899–918, 2020.

[60] T. Khalid, M. A. K. Abbasi, M. Zuraiz, A. N. Khan, M. Ali, R. W. Ahmad, J. J. Rodrigues, and M. Aslam, "A survey on privacy and access control schemes in fog computing," *International Journal of Communication Systems*, vol. 34, no. 2, p. e4181, 2021.

[61] J. Ling, J. Chen, J. Chen, and W. Gan, "Multiauthority attribute-based encryption with traceable and dynamic policy updating," *Security and Communication Networks*, vol. 2021, 2021.

[62] S. Ruj, A. Nayak, and I. Stojmenovic, "A security architecture for data aggregation and access control in smart grids," *arXiv preprint arXiv:1111.2619*, 2011.

[63] S. Ruj and A. Nayak, "A decentralized security framework for data aggregation and access control in smart grids," *IEEE transactions on smart grid*, vol. 4, no. 1, pp. 196–205, 2013.

[64] X. Fang, Q. Gan, and X. Wang, "Secure and efficient data aggregation scheme with fine-grained access control and verifiability for cwbans," *Journal of Internet Technology*, vol. 20, no. 3, pp. 771–780, 2019.

[65] W. Z. Khan, M. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, "Industrial internet of things: Recent advances, enabling technologies and open challenges," *Computers & Electrical Engineering*, vol. 81, p. 106522, 2020.