EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Fog computing network security based on resources management

Wided Ben Daoud[1*], Salwa Othmen[2], Monia Hamdi[3], Radhia Khdhir[4] and Habib Hamam[5,6,7,8,9]

*Correspondence:
wided.bendaoud@gmail.com

[1] NTS'Com Research Unit, ENET'Com, University of Sfax, Sfax, Tunisia
[2] Department of Computers and Information Technologies, College of Sciences and Arts Turaif, Northern Border University, Arar, Saudi Arabia
[3] Department of Information Technology, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, 11671 Riyadh, Saudi Arabia
[4] Department of Computer Science, College of Science and Arts in Qurayyat, Jouf University, Al-Jouf, Saudi Arabia
[5] Faculty of Engineering, Uni de Moncton, Moncton, NB E1A3E9, Canada
[6] International Institute of Technology and Management, Commune d'Akanda, P.O. Box 1989, Libreville, Gabon
[7] Spectrum of Knowledge Production & Skills Development, P.O. Box 3027, Sfax, Tunisia
[8] Department of Electrical and Electronic Engineering Science, School of Electrical Engineering, University of Johannesburg, Johannesburg 2006, South Africa
[9] College of Computer Science and Engineering, University of Ha'il, Ha'il 55476, Saudi Arabia

Abstract

Fog computing paradigm is designed as an extension of cloud computing due to the need for a supporting platform that is capable of providing the requirements of the Internet of Things (IoT). However, due to its features, fog obviously confronts numerous security and privacy risks, such as huge scale geolocation, heterogeneity, and mobility. Indeed, there are many problems resulting from security violations and breaches. Thus, to exceed these problems, we propose an efficient access control system, ameliorated with appropriate monitoring function and risk estimation to detect abnormal user's behavior and then deactivating illegitimate anomaly actions. Indeed, based on the risk value, we compute the trust level that will then be made into an access certificate, which would be provided to the user. This security certificate is used to authenticate and authorize users in case of re-connection in another time, without repeating the whole access control process from the beginning. Moreover, a comprehensive resource management mechanism is proposed to ameliorate the system performance and so to maintain low latency. Our aim is to further enhance data security, privacy and resource management for IoT users. To demonstrate the efficiency, feasibility, and security of our proposed scheme, we perform an extensive simulation using Network Security Simulator (Nessi2).

Keywords: Fog computing, Security, Access control, Risk, Trust, Access certificate

1 Introduction

Cloud computing is an effective platform that can handle the content in a distributed environment. However, this type of network is less efficient for applications that require mobility support, low latency, and remote zones with sensitive confidential applications and limited bandwidth. There upon, a novel platform called fog computing has been introduced in order to meet these requirements [1–4]. This new paradigm has been hosted in the edge of the network to serve the Internet of Things (IoT) devices with a best resources management and a lower delay. Certainly, fog systems do not replace the cloud; they are rather complementary to each other [5].

Accordingly, fog computing is a distributed computing platform that essentially extends the services delivered by the cloud system to the edge of the network, as shown in Fig. 1. Hence, fog computing is expected to have a basic and crucial role in the design of upcoming networks [6, 7].

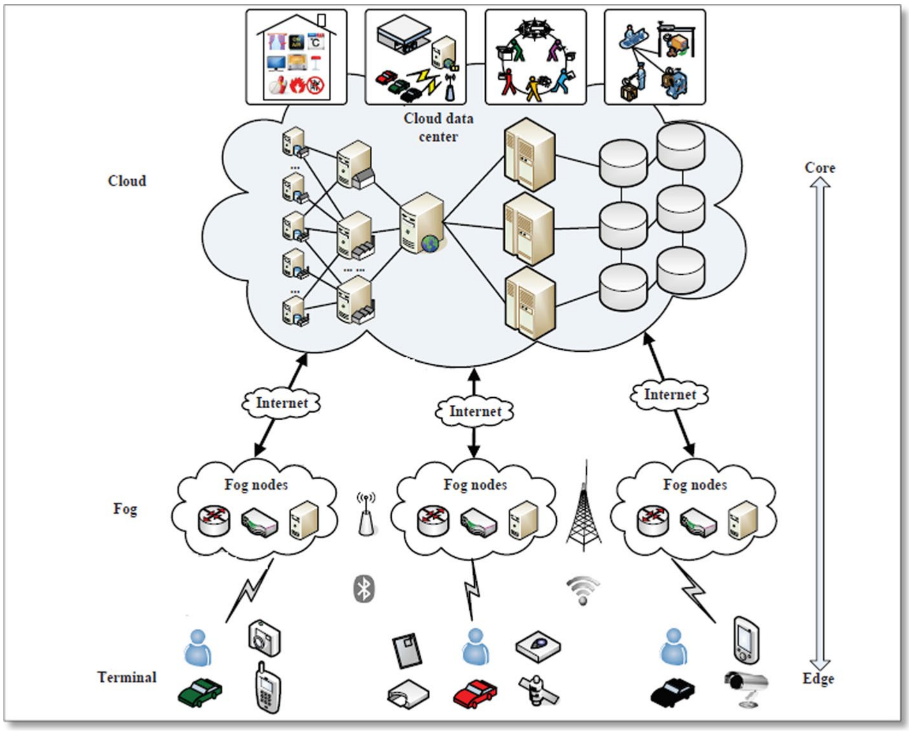


Fig. 1 IoT–fog–cloud computing architecture

To decrease latency, fog devices need to aggregate the data closer to the users. Consequently, when receiving data from different sources, the fog node decides whether to manage those data, using its own resources and services, or to send it to the cloud [6, 8, 9].

However, fog networks are facing serious and critical security challenges due to their inherent characteristics like heterogeneity as well as the more and more connected devices and the highly dynamic nature of these networks [8]. Moreover, some specific data need to be temporarily located in numerous fog nodes to facilitate computing, which makes the fog network more exposed to malicious attacks [10].

Furthermore, the fog platform is an attractive target for cybercriminals because of its high data transfer volume and its ability to acquire the data from the cloud and the IoT devices. Moreover, distributed fog computing nodes cannot be protected from complex attacks due to the shortage of global information from the entire network. Lastly, since fog nodes are distributed and scattered over large areas, the compromise of an unsecured edge node may be the breach point for an attacker [11].

Succinctly put, the sources of vulnerabilities in fog computing are numerous because it exists between the end IoT devices and the cloud data centers. By way of example, a hacker could deploy malicious applications, which could exploit a vulnerability that may damage or reduce the quality of service of the network [12].

Nonetheless, the existing strategies of security and defense of other types of networks are not adapted as they are inappropriate to the fog environment, because of the openness of the network. Likewise, traditional access control mechanisms are inefficient in this regard, so it is required to adopt an optimal and scalable solution for dealing with the limited resources of IoT devices.

The suggested mechanism must reduce the time taken to accept or reject a request considering that the execution time, the offload task time, the network search time, and the decision speed of the policy can all cause latency. This latency can in turn cause inflated expenses for the service provider. Delays in the process can have unwanted and adverse effects on the safety of people and the integrity of the infrastructure. Hence, it is essential to provide end users with guaranteed services and applications with reduced latency. In this respect, access control schemes in fog networks must grant access decisions within a lowest reasonable time.

To exceed some of the mentioned problems, an efficient security and quality management system, based on a set of security policies and several resource management techniques, is an important issue to enhance the intended security and performance of the system. In this respect, we propose a new approach to secure data in the fog cloud using a distributed strategy based on risk estimation. In addition, we recommend monitoring data access in the fog and detecting abnormal user's behavior and then deactivating illegitimate anomaly actions. Wherefore, our proposed approach is based on classification resources to succeed in a best management of services in fog environments. Moreover, based on the risk value, we compute the trust level that will then be made into an access certificate, which would be provided to the user. This security certificate is used to authenticate and authorize users in case of re-connection in another time, without repeating the whole access control process from the beginning.

The remainder of this paper proceeds as follows:

In Sect. 2, we describe and discuss security in fog computing. In Sect. 3, we detail the proposed access control architecture in fog computing. In Sect. 4, we present the obtained results of the evaluation of our model. Finally, the conclusions and perspectives are discussed in Sect. 7.

2 Related work

Fog computing has been presented and announced as a technology to narrow the gap between IoT devices and remote data centers. Nonetheless, fog devices obviously confront numerous security threats.

Many approaches have been proposed to enhance the security of data by using mechanisms like encryption or/and standard access controls and to discuss the policy-based management [13] to allow access to users in fog environments. Indeed, in [14], the authors proposed functional encryption schemes (FEs) in the continual leak memory model (CML). They developed a generic framework for the construction of leakage-resistant FE entities (LRFE) in the CML model that results from leakage-resilient pair encoding, which is the basic entity in the proposed framework. They adapted LRFEs by including FE for regular languages in addition to the attribute-based encryption (ABE) for large world and ABE with short cipher text. However, in [15], C. Mangla et al. proposed an approach to secure data transmission in fog computing using quantum cryptography which is a way used to provide a new computation method over classical computing. Indeed, quantum mechanics laws provide the power to quantum computing over the classical systems.

In [16], Qinlong Huang et al. proposed an access control model for IoT in fog computing based on ciphertext-policy attribute-based encryption (CP-ABE) and attribute-based signature (ABS). According to them, the sensitive information is encrypted and thus outsourced for cloud servers via neighboring fog nodes. The approved user decrypts the encrypted text stored in the cloud servers. Then, the legitimate user varies the decrypted data and re-outsources it with her/his signature. Accordingly, the cloud servers renew the ciphertext, given the condition that the attributes of the user in the signature suit the updated policy.

The previously presented works have been proposed to solve the problem of security in the fog computing environment. Nevertheless, fog systems are still facing certain challenges that need to be addressed.

In the field of resource management, numerous researches have been conducted addressing several issues with the aim of decreasing energy consumption, like in [17–20]. Additionally, in [14], Skarlat et al. proposed a peculiar definition in order to decrease network latency and attain a proper deployment of fog services. In this work, the researchers suggest a genetic algorithm as a solution for the fog service situation issue. In another effort, the authors in [21] proposed a solution for the fog service placement in a smart city situation through the use of a genetic algorithm. Moreover, Poltronieri, F. et al. proposed a novel scheme by using the adaptive, information-centric, and value-based (AIV) model for the service fabric management in order to better satisfy the requirements of users [22].

Furthermore, in [23], the authors created a fog development kit (FDK) which represents a realistic platform for the development of fog systems.

In [24], the authors made a study of monitoring system in the cloud computing based on their architecture, functions, and the properties. They present the different phases of the cloud monitoring processes and a comparative study of these phases. In [25], S. Chahida et al. proposed a new mechanism for risk assessment in IoT. In this mechanism, the authors identify the activities that must be protected and analyze different threats they face. The proposed mechanism is applied to IoT systems.

3 Proposed access control scheme

Our proposed approach is based on two factors: the available resources in order to succeed in a best management of services in fog environments, and the risk and trust evaluation that is updated during a monitoring process.

The mechanism of the risk estimation of user's behavior is established in this article to ensure the credibility of users. In fact, the trust evaluation in a fog environment cannot use authentication to decide whether the user is trusted or not [26]. The risk model that is centered on the user's credentials and behaviors is established in order to ensure the user's credibility and avoid threats imposed by her/his malicious activities to the fog nodes services. The proposed scheme quantifies information about the user, then collects feedback indicating the past behavior of the requester, and finally calculates the degree of the user's risk versus trust.

3.1 Components of the proposed scheme

The proposed model, shown in Fig. 2, presents the main process of access control where we have three basic functions, which are the risk assessment, the monitoring process, and the resource management:

- **Risk assessment:** is a process that quantifies the risk level associated with a specific threat or event, to improve the risk intelligence available to a system.
- **Monitoring process**: is a process of controlling and managing the operational workflow and processes within a cloud-based IT asset or infrastructure. The information collected by the monitoring is needed to ensure a correct execution of the cloud applications.
- **Resource management:** is a process that involves controls, timing, quality, availability, and general direction of resources development.

All these functions are achieved by the following components:

- **Local Agent Analyzer (LAA):** This agent is integrated into the fog layer. Its main role is to evaluate, in real time, all access requests with the user credentials. It asks for more information about her/his old access behavior to calculate the risk value associated with that user. Then, this component allocates an access certificate that indicates the trust level of that user.
- **Cooperator:** This component has the role of monitoring the entire network to observe the user's behavior. The cooperator collects all the information about users and then analyzes their behaviors and stores them in a related database. Furthermore, it provides a report that depicts the behavior of this consumer. In case of noticing a

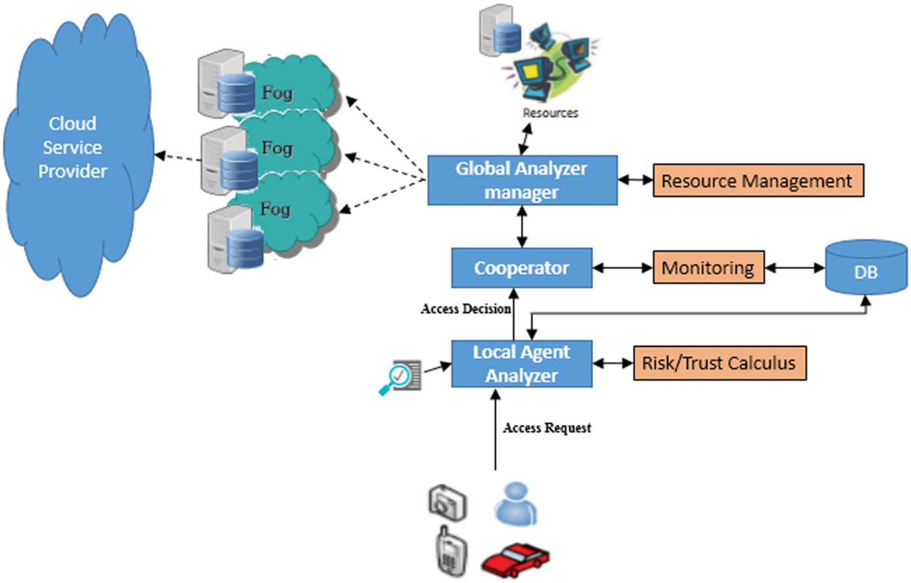


Fig. 2 Proposed access control approach

malicious activity, it acts in real time by stopping her/his actions (by rejecting the permission of that user) and subsequently avoiding damage.

- **Global Analyzer Manager (GAM):** It has a global view of the fog network. Therefore, its role is to allocate the requested resource to the requester, and in the case of a missing resource in the current fog node, this component must send queries to neighboring fog nodes to look for the requested service. Consequently, when a user sends her/his request for access, even if the requested resource exists locally, does not exist, or exists but with insufficient amount, this component facilitates the task. Hence, it avoids restarting the access procedure from the beginning, by searching again by itself in other neighboring fog nodes. The function of this manager manifests the basic characteristic of fog computing, which is ensuring a low latency.

3.2 Access control process

As shown in Fig. 3, when a user requests access to the fog networks, the Local Agent Analyzer (LAA) performs a global analysis, by including factors presenting the user identification data to be used in order to know whether that user is trusted or not. For instance, before accessing the services, the user must include attributes indicating her/his identity (ID), the requested resource, the place from where she/he requests access, and her/his function in the organization where she/he belongs (administrator, client). After that, these attributes are used to calculate the risk value associated with that user. Then, this value is transmitted to the cooperator component that monitors the entire network and saves the information associated with each user's behavior in the system.

Based on the risk analysis result, the user is allowed or denied access to the fog/cloud resources. The granted permission represents a set of privileges, which are required to access the requested services. In the optimal case, when the credibility value of the user is important and the demanded resources are available, the fog system should allocate the maximum of the requested resource to that user, when it is possible. However, if the

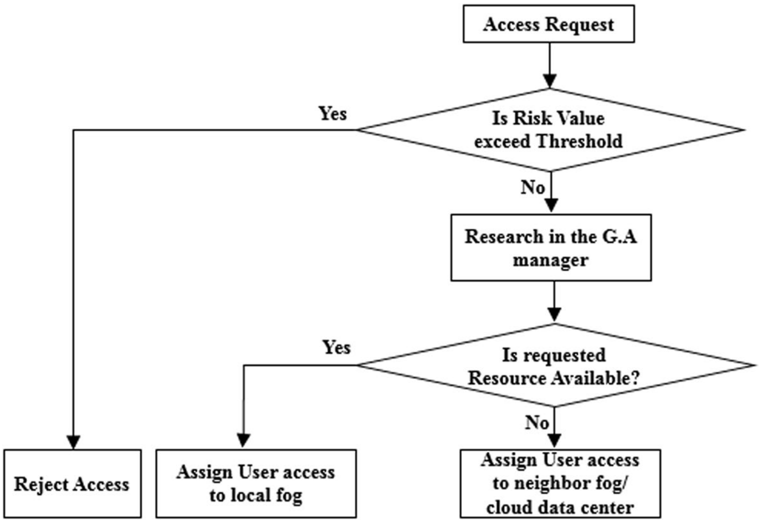

Access Request
Is Risk Value exceed Threshold
Yes
No
Research in the G.A manager
Is requested Resource Available?
Yes
No
Reject Access
Assign User access to local fog
Assign User access to neighbor fog/ cloud data center

Fig. 3 Proposed access control process

credibility value is very low, the user might receive a resource lower than the requested privileges.

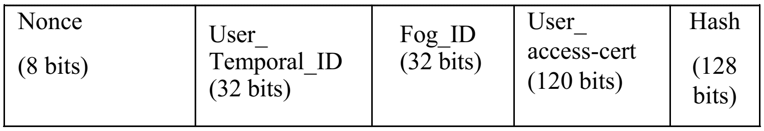
When granting permission to access the network, a trust certificate is delivered to the user. This certificate is to be used for the next access.

If the requested resource exists locally (in the current fog node), the GAM allocates the demanded resource to the user. If the resource does not exist locally, and since the GAM has a global view of the fog cloud network, it would send queries to neighboring fog nodes to look for the requested service. If the search procedure is successful, the found resource is allocated to the user. Yet, if the user changes her/his attitude and acts as an attacker, the cooperator acts actively by disabling her/his access permission and then stopping the actions.

Such access rules are configured into the control mechanisms using a policy language. A predominant standard used for this purpose is eXtensible Access Control Markup Language (XACML) [27], which is an XML-based language that supports fine-grained access control. The XACML language permits to produce flexible policies to define the requirements for retrieving resources. This standard describes both a policy language and an access control decision response language. The XACML supports a wide variety of data types, functions, and rules about combining the results of different policies into one decision. In this regard, we are using the XACML message in order to present the response of user's access decision, delivered from the fog node to that user. The format of the adapted XACML message is shown in Fig. 4.

Here,

- **Nonce (8 bits)** is a random number that is used only one time. It is used to ensure the anti-replay security service.
- **User_ Temporal_ ID (32 bits)** is the temporal identity of a user, who has requested some services. This field is changed in every connection session established between the user and the fog network.
- **Fog_ID (32 bits)** is the identity of the fog node from which the access response will be sent. Each fog node will have an ID that will be shared between fog nodes in order to know the delivered services that form each one. This helps to ensure the proper resource management in our system.
- **User_access-cert (120 bits)** is the user access certificate, delivered by the GAM. It contains essentially the trust level of that user.
- **Hash (120 bits)** is the hash function of the previously presented fields. In the phase of access decision, the message is hashed before being sent to the user. This ensures that the access process is not susceptible to extension attacks that can be added to the message and which can cause enormous threats in the resource information. The mechanism of hashing is realized as illustrated in function (1):



Nonce (8 bits)	User_ Temporal_ID (32 bits)	Fog_ID (32 bits)	User_ access-cert (120 bits)	Hash (128 bits)

Fig.4 Format of exchanged message

$$ID_t = h(User\_Temporal\_ID|Nonce|fog\_ID) \tag{1}$$

where h is a hash function (HMAC).

3.3 Risk calculus and analysis

Our proposal is mainly based on a risk value which includes a set of factors that are measured for every access request and aggregated to determine the total security risk. Some examples are presented in Table 1. This value is computed based on formula (2),.

Here, W_i is the *weight* attributed to metric i and x_j is its risk value.

Risk are linear combinations of inputs x_j.

$$\text{Risk} = \sum_i W_i x_i \tag{2}$$

To improve our knowledge of the risk assessment and to retrieve the adjusted values of weights, we present a mathematical model using a neuron application of artificial intelligence, where the multilayer perceptron (MLP) neural network is proposed.

As illustrated in Fig. 5, the MLP neural network is divided into three layers: the input layer, the hidden layer, and the output layer. The risk computation consists of an input vector $X = x_1, x_2, \ldots x_n$, at the input layer; a weight vector $W = w_1, w_2, \ldots w_n$; a summation block that combines the inputs and weights at the output layer; and an activation function f that, depending on the risk value and a threshold (chosen randomly and dynamically), makes decision as an output function.

The selection of the input of our model is based on risk factors. For each situation, the risk assessment process collects information about the risks and failure of the system or the network, which results in a new risk factor.

The result of applying the weights $W_{i(i=1\ldots m])}$ to the inputs $x_{j(j=1\ldots n])}$ is the *Risk* value. The function f is defined as follows:

$$f(Risk) = \begin{cases} 1 & if\ risk \geq riskThreshold \\ 0 & Otherwise \end{cases} \tag{3}$$

Table 1 Simulation parameters

Parameter	Value
Simulator	Nessi2
Host	Windows 7, 64bits
Tick	0.05 s
Simulation time	1000tick
Channel type	Wireless channel
Traffic type	TCP
Data payload	529 bytes/packet
Number of fog nodes	4
Number of connected users	2-5-7-10-13-18-20
Frequency	2.16 GHz
Data rate	11.4 Mbps

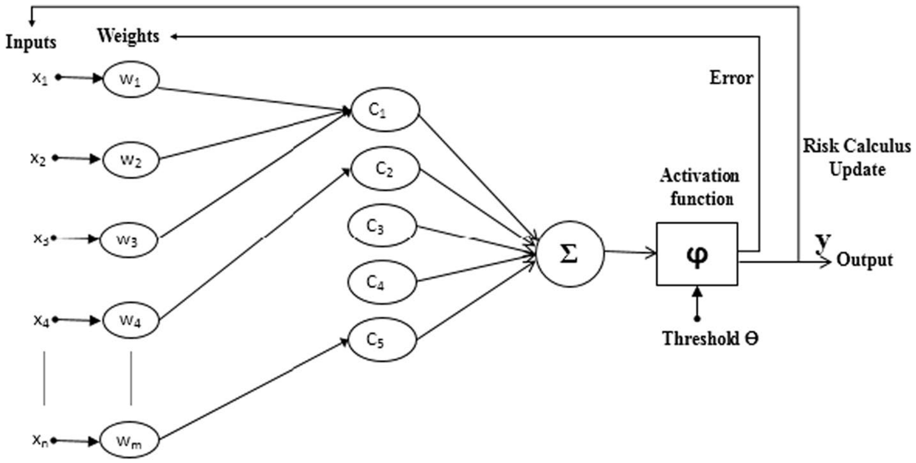


Fig. 5 Used neural network

The *risk Threshold* is applied to the value *Risk* to produce an output value, OUT = f(*Risk*).

After this procedure, we apply the trust evaluation to our model. If the trust value of a user decreases due to any abnormal behavior detection or any unfulfilled obligation in the system, while that user has an open session, the manager sends a notification to re-evaluate the risk metrics in order to re-calculate the *risk* value. This allows the system to deny access to misbehaving users before they can do significant damage to the system. In such a case, the user's permissions are automatically deactivated. Given that the trust level is inversely closely dependent on the risk value, we propose to calculate the trust level using Eq. (4).

$$Trust = 1/Risk \tag{4}$$

Subsequent to the risk and the trust assessment, the system will deliver an access certificate that would be used by the client in order to be authenticated by other fog/cloud service providers. In such a case, the client does not need to repeat the whole access control procedure. Owing to this certificate indicating her/his trust level, the user can re-connect to fog nodes without going through the risk analysis phase. The access certificate content is given by:

$$\begin{aligned} access\_cert = \{ & \boldsymbol{Trust\ level}, \boldsymbol{ID\_User}, \boldsymbol{ID\_Fog}, \boldsymbol{Access\ period}, \\ & \boldsymbol{Resource\_allocated}, \boldsymbol{Manager's\ Signature} \} \end{aligned}$$

where **Trust level** is a trust value calculated based on the risk calculus. **ID_User** is the identity of the user who demands the access to the fog node. **ID_Fog** is the ID of the fog node delivering a required service and a certificate. **Access period** is the GAM that restricts the period of access to such a service in the system. **Resource_allocated** is the resource allocated to the authorized user. **Manager's Signature** is the signature assigned by the manager to the certificate.

We notice that the certificate is generated by the GAM component in concordance with the LAA, and it is revoked, if the risk value is important, to refuse the user access.

Based on the model network shown in Fig. 5, we use the supervised training method by which, for each weight vector W and input vector X applied to the network, the result is compared to the target and then an error is calculated. This error is deployed to adjust the weights by minimizing the error, defined by formula (7). The MLP neural network is characterized as follows:

- Output y are linear combinations of inputs x_i

$$\mathrm{y} = \sum_{i} \omega_i x_i \tag{5}$$

- Error function for a particular input n is

$$E_n = \frac{1}{2} \sum_{k} (y_n - t_n)^2 \tag{6}$$

where $y_n = y(x_n, \omega)$

- Gradient of Error function wrt a weight w_{ji}:

$$\frac{\partial E_n}{\partial \omega_{ji}} = \left(y_{nj} - t_{nj}\right) x_{ni} \tag{7}$$

- Local computation involving product of error signal y_{nj}-t_{nj} associated with output end of link w_{ji}
- x_{ni} measures are associated with the input end of the link

For a particular input x and weight w, squared error is:

$$E = \frac{1}{2} \left(y(x, \omega) - \mathrm{t}\right)^2 \tag{8}$$

$$\frac{\partial E}{\partial \omega} \left(y(x, \omega) - t\right) x = \delta . x \tag{9}$$

$$\frac{\partial E}{\partial \omega_{ji}} = \left(y_j - t_j\right) x_i = \delta_j x_i \tag{10}$$

3.4 Monitoring and Risk Update

The monitoring task is an important function in our proposed system, which is performed by the cooperator. The system calls this function continuously during the life cycle of the session of the user. The activities and the behaviors of each user are maintained in a log file. After requesting the access, the cooperator submits the user's feedback report, which is used to update the risk value. The risk is updated based on i) detected abnormal factors ε from the user's feedback report and ii) updated errors from the MLP neural network presented in Fig. 5.

Table 2 Risk factors and associated weights

Risk factor	Weight	Risk factor	Weight
Characteristics of Requester	**20**	**Characteristics of The Information Requested**	**20**
Role	5	Classification Level	5
Rank	5	Encryption Level	5
Clearance Level	5	Network Classification Level	5
Education Level	5	Permission Level	5
Characteristics of IT Components	**20**	**Situational Factors**	**20**
Machine Type	3.3	Specific Mission Role	3.3
Application	3.3	Time Sensitivity of Inf	3.3
Connection Type	3.3	Transaction Type	3.3
Authentication Type	3.3	Current Location	3.3
Network	3.3	Threat Level	3.3
Distance between requester and source	3.3	Physical location of the requester	3.3
Heuristics	**20**		
Risk Knowledge	5		
Trust Level	5		
Previous Violations	5		
Previous Access	5		

$$\text{Risk} = \sum WiXj + \varepsilon + E_n \tag{11}$$

where $\varepsilon = g\left(risk\_factors\right)$

$$g\left(risk\_factors\right) = \sum (Wi + w_e)(Xj + x_m) \tag{12}$$

wherew_e is the "new" adjusted weights that are re-calculated after the back-propagation of the error function and x_m is the updated risk factors input, which represent the detected abnormal factors.

The function $g\left(risk\_factors\right)$ depends on the return results from the monitoring system, which returns results based on abnormal factors for that user.

The details about metrics will be presented later in Table 2 [27, 28].

Accordingly, when the monitoring system detects some changes in the user's behavior, the main system should modify the values of the factors in question. On successful detection, the system lowers the risk threshold to stop certain activities. It then calls an automated deactivation function to deactivate certain permissions.

3.5 Resource Management

Fog computing should allow great access control techniques over data and network with reasonable resource allocation strategies to safeguard integrity and confidentiality in the multi-user ecosystem.

Based on performance requirements and resource restrictions, a resource management plan is produced using the designed classification mechanism. The function of resource classification is to classify different IoT services permitted by the different fog nodes that are existing and idle in the network at that time. The GAM must send queries to neighboring fog nodes to know the service delivered by each node. At least, each fog

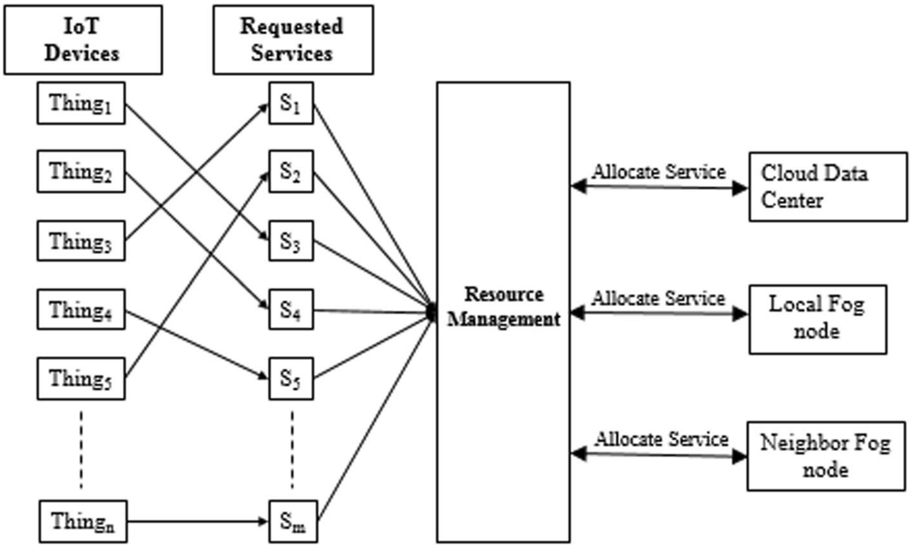


Fig. 6 Operating architecture

node should have a database, which contains the properties of its neighboring fog nodes. These records are updated continuously. When fog nodes are highly utilized, they do not satisfy all services requests. This new procedure is created to provide access to various IoT devices, when the resources of fogs do not suffice to provide services.

Figure 6 presents the operating architecture, where the fog resources are scheduled by service type. The requested services are served using fog or cloud system based on the predefined resource classification mechanism.

4 Performance evaluation

The suggested access control scheme is evaluated in terms of theoretical and experimental effects, which show the efficiency of the proposed model in terms of security and quality of service (QoS).

4.1 Simulation experiments

4.1.1 Simulation description

We investigate the performance of our proposal for cloud fog computing networks via simulation analysis using the Network Security Simulator (*Nessi2*). We implement the architecture of our network solution using this simulator. The simulation parameters are presented in Table 1.

To evaluate our proposal, as a first step, we create the network topology. Then, we implement our algorithms using the "developer-studio-eclipse-jee" as an editor to develop the proposed policies. Figure 7 shows the architecture of the simulated scheme.

NB: Nessi2 offers the ability to examine network protocols like TCP and UDP.

As shown in Fig. 7, we created four access subnets representing "IoT users," which contain a various number of users. Besides, we made four fog nodes containing different types of servers. These fog nodes are near to users in such a way that every time the

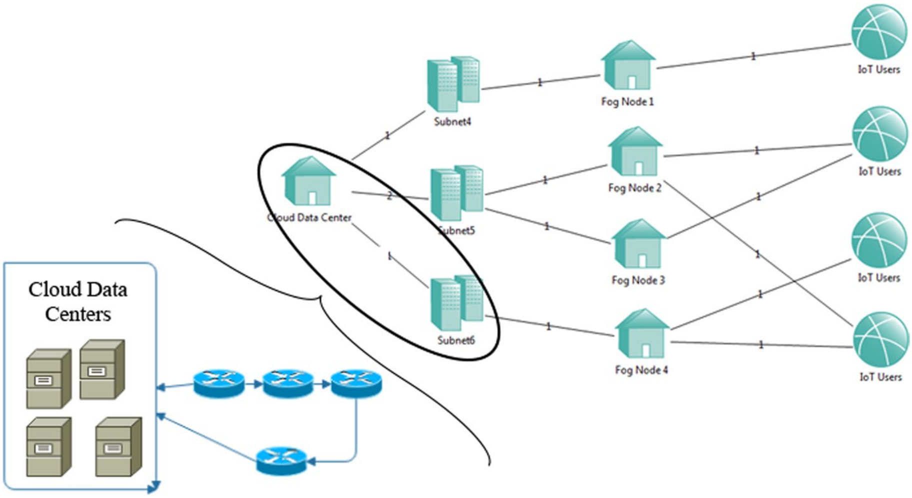


Fig. 7 Simulation architecture

user requests services, she/he can connect very quickly to these nodes, unlike the case for cloud. The requests are then processed using routers to find the cloud data centers, which are composed of a large number of servers. Routers are responsible for the connection between the elements of our simulated architecture.

Before giving the results in terms of execution time and throughput, we discuss in the following sub subsection the risk and trust calculus used in the simulation.

4.1.2 Risk and trust calculus

In this section, we detail our approach by giving examples on the risk and trust assessment. We have seven users, and for each one of these users, we calculate the risk in order to know the trust level, and accordingly, the access decision that can either be 'allowed' or 'denied' is given.

Based on [27], Table 2 presents the risk factors used for the simulation done based on our model. These factors are measured for every access request and aggregated to determine the total security risk.

- Category 1 (Characteristics of the Requester) is the risks associated with the user or the application that request an access to a specific resource.
- Category 2 (Characteristics of IT Components) relates to the risks which are assigned to the components in the path between the requester and the resource, like the type of application, machines, and network.
- Category 3 (Situational Factors) are risks assigned to the situation surrounding the request, like the role of the requester to perform a specific task and the critical time of the requested resource. Additionally, it includes the risks associated with the environment surrounding the request, like the physical location of the requester.
- Category 4 (Characteristics of the Information Requested) is the risks assigned to the resource, like the classification, permission, and encryption levels.

- Category 5 (Heuristics) is the risk associated with previous similar requests, like known violations (risk knowledge) and successful transactions (trust level).

Each category has a total weight of 20 (100/5), and each factor in a category has a weight of 20/n, where n is the number of metrics in that category.

Below, we give an example of the role metric:

$$Risk_{Role} = \left\{ \begin{matrix} 1 if user \in SuperAdmin \\ 5 if user \in Admin \\ 10 if user \in User \\ 15 otherwise \end{matrix} \right\} \tag{13}$$

The values presented in (13) are just an example of simulation and dissimilar systems can deploy different values.

As a matter of fact, our model is interesting in light of the fact that it combines various metrics that were separately mentioned in a number of previous works like in [27]. It also illustrates the feasibility of having a numerous set of factors that are applicable in various systems as it demonstrates how important it is to have a mechanism that is appropriate for a varying set of metrics. Thus, that system can introduce its policies depending on its own applications. Therefore, it is easy to envisage many custom systems, which can simply deploy this model by removing or adding a new group with its own set of metrics.

4.2 Simulation results

The performance of our solution is established by the results that will be presented below. In order that we measure the delay and the throughput produced along the exchange of data between users and servers, we compare the two performance metrics listed below for cloud and fog.

- **Average execution time**: is the time required to grant permission to certain services and to receive the requested resources.

$$\text{Average execution time} = \text{reception time} - \text{transmission time} \tag{14}$$

- **Throughput**: refers to the measure of the average data rate that the system can process in a specified amount of time.

$$P = \sum_{i=1}^{N} V_i \Big/ T$$

where V_i presents the number of packets received by the network device i and T indicates the total time elapsed during the communication and data exchange between two nodes.

(A) Average Execution Time

Figure 8.

(B) Effect of the average throughput

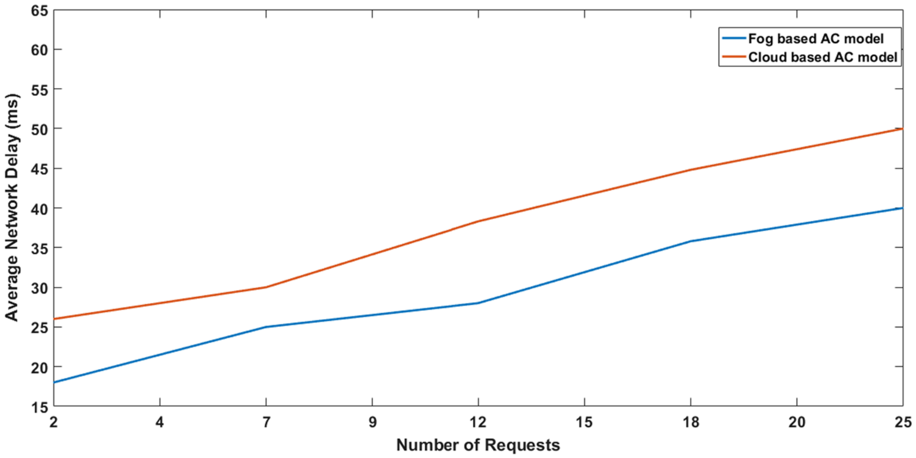


Fig. 8 Comparison of system execution time

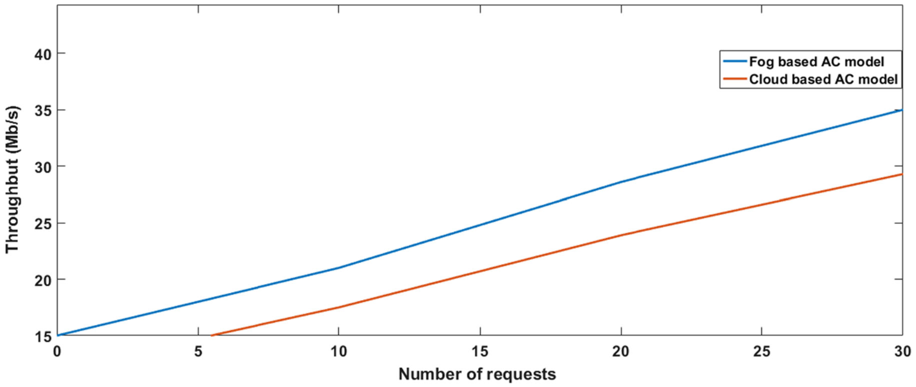


Fig. 9 Comparison of system access throughput

In this experiment, we calculate the throughput overall data exchange while employing our solution in fog system and then in the cloud.

The results, depicted in Fig. 9, illustrate that our access control strategy employed in the fog achieves the highest average throughput. In fact, for higher numbers of dispersed services, the allocation result in a lower delay, thus the throughput will be increased. Furthermore, this is explained by the fact that, with the growing number of connected devices, there is a big data flow generated at an exponential rate. All these amounts of data, routed to the edge of the network or to the remote cloud data center, would encumber the network. Hence, offloading all actions to the cloud is not a practical option that requires the handling of data by the fog nodes. Then, the deployment of our model, which ensures the resource management, allows the optimization of the resources between consumers.

4.3 Achieved security issues

The fog cloud concept requires the usage of the following security services in a coordinated way. In this regard, we analyze the security features that are expected to be achieved by the proposed model.

- ***Confidentiality*****:** In our proposed scheme, the model evaluates the risk values based on the user attributes or credentials to verify the identity of each end user device and then, to ensure the confidentiality of the data stored in fog/cloud servers. After that, it decides to permit or deny the access request, based on a trust certificate indicating the trust level of the user, before damaging the system.
- ***Availability*****:** In our proposed security model, we approve the system availability after employing the access control mechanism. This new procedure is created to provide access to various IoT devices when the resources of fogs do not suffice to provide services. Therefore, there will be no shortage of resources, and hence availability of all services will be granted.
- ***Anti-replay:*** We use "Nonce" as a random number sent only once to guarantee the anti-reply of the response message containing the user's identifier from an attacker.
- ***Identification/ Authentication:*** During registration, when a user requests a given service, she/he will have not only a *User_ID* but also a *User_temporal_ID*, which is a temporal ID generated for each session. In addition, our scheme exploits the hash function to achieve anonymity. Thus, we invoke the notion of access certificate, which indicates the authentication of that user, and which also authenticates the network entity that delivers and evaluates the trust level.
- ***Intrusion Detection and Prevention****:* A monitoring process is realized continuously, even after the user gains permission. Beside, a deactivating function is triggered when detecting abnormal activities that would then be reported.

5 Conclusion

Data security and privacy represents a critical issue for all paradigms that use big data. In this paper, an efficient access control model with risk and trust evaluation capabilities for fog computing is proposed to ensure the privacy and the security in fog environment. Indeed, our proposed approach is based on a resource classification method to succeed in a best management of services in fog networks and then to address the issue of data leakages. Data in fog computing are very critics, and so, their divulgation by an attacker may result a catastrophic problem; our proposed approach is an optimal solution to prevent these critical data from the attackers. Indeed, we propose a mechanism to detect the abnormal user's behavior and then deactivate illegitimate anomaly actions. To succeed in a best management of service in fog environments, our proposed approach takes into consideration the classification resources.

Then, the performance of the proposed access control mechanism is evaluated, and the results show that the proposed model gives lower delay with higher security.

As a future work, we plan to develop the certificate manager component for fog computing networks, which can manage access certificates to users.

6 Methods/experimental

In this study, the authors propose a mechanism to manage the resource in fog computing based on the risk and trust evaluation capabilities. Our aim is to ameliorate the security of the exchanged data and ensure the privacy in fog environment and so improve its performance.

To demonstrate the efficiency, feasibility, and security of our proposed scheme, we perform an extensive simulation using Network Security Simulator (Nessi2) by creating the network topology. Then, we implement our algorithms using the "developer-studio-eclipse-jee" as an editor to develop the proposed policies.

Abbreviations

GAM	Global Analyzer Manager
IoT	Internet of Things
LAA	Local Agent Analyzer
Nessi2	Network Security Simulator
XACML	EXtensible Access Control Markup Language

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU-FFR-2023-0061".

Author contributions

WBD analyzed and interpreted the fog security issues. WBD, SO, and MH had contributed to propose the final solution in the article. SO performed the related works and was a major contributor in writing the manuscript. RK was responsible for final manuscript editing. HH and MH were responsible for funding acquisition. All authors read and approved the final manuscript.

Funding

This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2023R125), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Availability of data and materials

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 23 February 2023 Accepted: 1 June 2023
Published: 14 June 2023

References

1. A. Karakaya, S. Akleylek, A novel IoT-based health and tactical analysis model with fog computing. PeerJ Comput. Sci. **7**, 1–34 (2021)
2. G. Caiza, M. Saeteros, W. Oñate, M.V. Garcia, Fog computing at industrial level, architecture, latency, energy, and security: a review. Heliyon **6**, e03706 (2020)
3. S. Mahfoudhi, M. Frehat, Enhancing cloud of things performance by avoiding unnecessary data through artificial intelligence tools, in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 1463–1467 (2019)
4. M. Al-Masarweh, T. Alwada, W. Afandi, Fog computing, cloud computing and IoT environment: advanced broker management system. J. Sens. Actuator Netw. **11**(84), 1–17 (2022)
5. W. Saeed, Z. Ahmad, A.I. Jehangiri, N. Mohamed, A.I. Umar, A Fault tolerant data management scheme for healthcare internet of things in fog computing. KSII Trans. Internet Inf. Syst. **15**(1), 35–57 (2021)
6. V.H. Osmanaj, A. Al-ahmad, Fog computing security and privacy for the Internet of Thing applications: State-of-the-art. Security Privacy **4**(e145), 1–26 (2021)
7. T. Khalid, M. Abbas, K. Abbasi, M. Zuraiz, M. Aslam, A survey on privacy and access control schemes in fog computing. Int. J. Commun. Syst. **34**(e4181), 1–39 (2021)
8. N.N. Khumalo, L. Mfupe, O.O. Oyerinde, Reinforcement learning-based resource management model for fog radio access network architectures in 5G. IEEE Access **9**(3051695), 12706–12716 (2021)
9. B.S. Khater et al., applied sciences A lightweight perceptron-based intrusion detection system for fog computing. Appl. Sci. **9**(178), 1–21 (2019)

10. W. Ben Daoud, M. Rekik, A. Meddeb-makhlouf, F. Zarai, S. Mahfoudhi, SACP : Secure Access Control Protocol, pp. 935–941 (2021)
11. Q. Duy, M.V. Ngo, T. Quang, T.Q.S. Quek, H. Shin, Enabling intelligence in fog computing to achieve energy and latency reduction. Digital Commun. Netw. **5**(1), 3–9 (2019)
12. D. W. B. I. A. Brown, A Security risk measurement for the radac model, no. March (2007)
13. C. Dsouza, G. Ahn, Policy-driven security management for fog computing: preliminary framework and a case study, in *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, pp. 16–23 (2014)
14. Z. Yu, M. Ho, Q. Xu, R. Yang, J. Han, Towards leakage-resilient fine-grained access control in fog computing, *Future Generation Computer Systems*, pp. 1–15 (2017)
15. C. Mangla, S. Rani, H.K. Atiglah, Secure data transmission using quantum cryptography in fog computing. Wirel. Commun. Mob. Comput. **2022**, 1–12 (2022)
16. Q. Huang, Y. Yang, L. Wang, Secure data access control with ciphertext update and computation outsourcing in fog computing for internet of things, vol. 5 (2017)
17. S. Saraswat, S. Member, H.P. Gupta, Energy efficient data forwarding scheme in fog-based ubiquitous system with deadline constraints. IEEE Trans. Netw. Serv. Manag. **17**(1), 213–226 (2020)
18. Z. Lin, M. Lin, B. Champagne, W.P. Zhu, N. Al-Dhahir, Secrecy-energy efficient hybrid beamforming for satellite-terrestrial integrated networks. IEEE Trans. Commun. **69**(9), 6345–6360 (2021)
19. Z. Lin, K. An, H. Niu, Y. Hu, S. Chatzinotas, G. Zheng, J. Wang, SLNR-based secure energy efficient beamforming in multibeam satellite systems. IEEE Trans. Aerosp. Electron. Syst. (2022). https://doi.org/10.1109/TAES.2022.3190238
20. Z. Lin, M. Lin, T. De Cola, J.B. Wang, W.P. Zhu, J. Cheng, Supporting IoT with rate-splitting multiple access in satellite and aerial-integrated networks. IEEE Internet Things J. **8**(14), 11123–11134 (2021)
21. C. Canali, R. Lancellotti, A fog computing service placement for smart cities based on genetic algorithms, no. Closer, pp. 81–89 (2019)
22. F. Poltronieri, M. Tortonesi, A. Morelli, C. Stefanelli, N. Suri, Value of information based optimal service fabric management for fog computing. IEEE/IFIP Network Operations and Management Symposium, pp. 1–9 (2020)
23. C. Powell, C. Desiniotis, B. Dezfouli, The Fog development kit: a platform for the development and management of Fog systems. IEEE Internet Things J. **7**(4), 3198–3213 (2020)
24. M. N. Birge, C. Bulla, Cloud monitoring system: basics, phases and challenges. Int. J. Recent Technol. Eng. (IJRTE) 8(3) (2019)
25. S. Chahida, A. Bayoua, P. E. Brun, M. Cantera, Risk Assessment in IoT Case Study: Collaborative Robots System, Creative Commons License Attribution 4.0 International, pp. 3–10 (2020)
26. N. W. Group, The Intrusion Detection Message Exchange Format (IDMEF) (2007)
27. W. Ben Daoud, A. Meddeb-Makhlouf, F. Zarai, A model of role-risk based intrusion prevention for cloud environment, in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, pp. 530–535 (2018)
28. D.R. Dos Santos, R. Marinho, G.R. Schmitt, C.M. Westphall, A framework and risk assessment approaches for risk-based access control in the cloud. J. Netw. Comput. Appl. **74**, 86–97 (2016)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.