A Distributed Access Control Scheme based on Risk and Trust for Fog-cloud Environments

Wided Ben Daoud[1], Amel Meddeb-Makhlouf[1], Faouzi Zarai[1], Mohammad. S. Obaidat[2]
and Kuei-Fang Hsiao[3]
[1]*NTS'Com Research Unit, ENET'COM, University of Sfax, Sfax, Tunisia*
[2]*Department of ECE, Nazarbayev University, Astana, Kazakhstan, University of Jordan, Amman, Jordan*
[3]*Ming-Chuan University, Taiwan*

Keywords: Fog Computing, Access Control, Risk, Trust, Resource Manager, Monitoring.

Abstract: Fog computing is a technology, which benefits from both Cloud Computing and Internet of Things paradigms. Instead of centralizing the information stored in the cloud, the idea in the fog environment is to use devices located at the edge of the network to prevent congestion during data exchanges. Since its development, the fog computing was integrated in numerous applications, including, connected vehicles (v2x), and smart cities, among others. Furthermore, the fog environment is highly variable within the resources as it varies dynamically. Therefore, a careful management of resources is essential for maximizing the efficiency of this distributed environment. That is why, the access control to resources is one of the challenging issues to be taken into account to enhance security in fog environments. In this paper, we develop a novel distributed fog-cloud computing mechanism for access control based on trust and risk estimation. We evaluate the performance of the novel approach through appropriate simulations, which improves system performance by minimizing the time spent to have permissions to access services and optimizing the overall system resource utilization.

1 INTRODUCTION

There are recent technologies that are increasingly making our lives more comfortable and more convenient. Cloud computing is a good example on these. It simplifies the storage and the management of data with higher efficiency and lower cost. Another new technology is the Internet of Things (IoT), introducing and changing our daily lives is the Internet of Things (IoT) (Almutairi et al., 2012), (dos Santos et al., 2016), which connects physical objects and enables these objects to collect and exchange data.

Fog computing combines the benefit of both the cloud computing and the Internet of Things. In brief, the concept of fog computing is an extension of the paradigm of cloud computing to the edge of the network. A scalable and elastic fog computing environment is intrinsically very dynamic. Thus, the major advantage of using this new technology lies in its capacity to not overload the information network. Consequently, users can store their data online in remote servers and access them from anywhere (dos Santos et al., 2016). This makes the problem of providing adequate access control in fog even more difficult. In addition, as fog computing offers new opportunities to exchange personal data between different fog nodes and cloud service providers, where security and privacy experience critical issues in the expansion of any sensitive information. Given that this paradigm is an open platform and the interaction between participants among themselves in a fog environment is high, this can cause various security concerns. Clearly, there is a need to protect the privacy of data from malicious attacks. Therefore, we propose a novel distributed model of access control to improve the existing mechanisms and, especially to facilitate traffic management in fog computing environment.

In this paper, we examine in more details the access control mechanism for fog-cloud environments. Our goal is to propose a comprehensive model to control user access to the fog computing environment. The purpose behind our work is to ensure not only the privacy and the safety of shared information, but also to reduce the

Daoud, W., Meddeb-Makhlouf, A., Zarai, F., Obaidat, M. and Hsiao, K-F.
A Distributed Access Control Scheme based on Risk and Trust for Fog-cloud Environments.
DOI: 10.5220/0006848502960302
In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018) - Volume 1: DCNET, ICE-B, OPTICS, SIGMAP and WINSYS*, pages 296-302
ISBN: 978-989-758-319-3

complexity of administration and management of security mechanisms.

1.1 Need for Distributed Access Control in the Fog-cloud

In a distributed environment such as fog computing, access control issues pose serious challenges to the widespread adoption of cloud-based IoT services. Thus, the short distance between the user and the information also offers greater efficiency and requires more security. The fog computing is a dynamic paradigm, which makes resources distributed and pervasive. Therefore, we require a judicious administration of resources in order to enhance the effectiveness of the fog.

More importantly, the cloud-fog cooperation will produce an enormous potential. Hence, to decrease heavy computational and communication overhead on service providers or data owners, we propose a distributed architecture of access control.

1.2 Brief Description of Fog Computing

The Fog Computing model is designed as an extension of the cloud. The term "fog" was initially proposed by Cisco to designate the need for a supporting platform capable of ensuring the requirements of IoT (Huang et al., 2017), (Shirazi et al., 2017). Fog computing is described as a highly virtualized platform that delivers storage, computing and networking services between the terminals and traditional cloud computing data centers, at the edge of the network (Figure. 1). Thus, fog computing supports cloud computing to be more evolving and scalable (Hu et al., 2017).

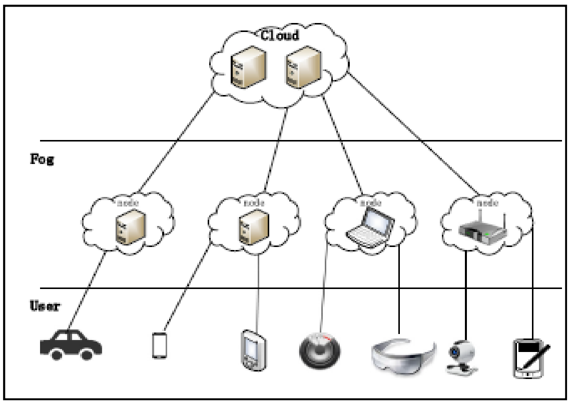


Figure 1: The concept of Fog computing.

This paradigm is generally deployed in applications requiring low latency. Actually, cloud can easily host many fog applications. Table 1 presents the fog computing characteristics.

Table 1: Cloud and Fog characteristics requirements.

Requirements	Cloud	Fog
Latency	High	Low
Distance	Through Internet	Limited Hop Count
Bandwidth	More Demand	Less Demand
Mobility	Limited	Supported

In fog computing, privacy of data is a requirement that must be met when analyzing sensitive information. In support of this requirement, optimal and secured access control processes should be considered. Besides, devices, as well as the service providers are heterogeneous and distributed, which may increase problems in managing resources and the user's access to these resources.

Subsequently, the application of security in fog environments is a major challenge. The goal of access control for services is to maximize the efficiency of resource utilization, and satisfy the users' security and privacy needs, meanwhile, take full advantage of the profit of both fog providers and cloud service providers (Shirazi et al., 2017), (Dastjerdi and Buyya, 2016).

The rest of this paper is organized as follows. Section 2 describes a trust-risk aware approach for security improving access control scheme. Then, an experimental evaluation of the proposed solution in terms of performance will be discussed in Section 3. Section 4 concludes the paper and provides concluding remarks.

2 OVERVIEW OF THE PROPOSED SOLUTION

To manage the scalability problem in terms of users and resources, we propose a distributed access control architecture in fog-cloud computing. Each fog node or cloud server applies independent access control policies, and it is necessary to have mutual access to resources between domains, which implicates a corresponding access control mechanism to manage the interoperability inter-domain (Almutairi et al., 2012). The proposed distributed architecture, which is summarized in Figure. 2, processes and integrates the risk and the trust concepts. Then, it is constructed using three types of components, as shown in Figure. 4: Resources Manager (RM), Distributed Access

Control Administrator (ACA) and Service Level Agreement (SLA).

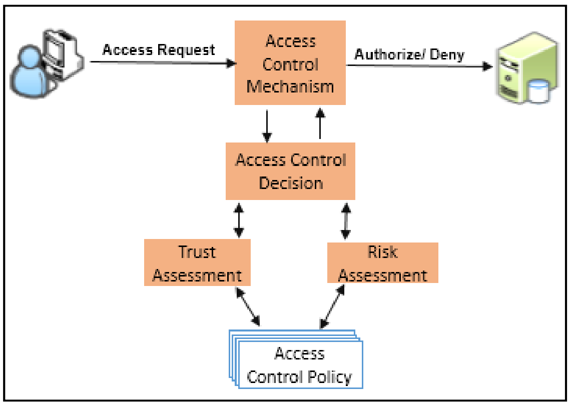


Figure 2: The main strategy proposed for access control.

2.1 Risk and Trust based Access Control Definitions

2.1.1 Risk

Risk estimation is one of the basis of the proposed technique. It is defined by the likelihood of a dangerous and harmful circumstance and its consequences if it happens. All research has the ultimate objective to reduce the possibility of occurrence of a certain risk through the application and the implementation of mechanisms of control and policies in the system. Therefore, the techniques of risk estimation are required to ensure an automatic prevention and protection from insider attacks.

Currently, there are risk factors, which represent all possible outcomes that can be found about a costumer. The risk value is calculated after finding the probability of each event, which is then multiplied by a weight attributed by experts to each metric. These factors are evaluated for every access request and aggregated to achieve a measure of the total security risk. Details about all the metrics can be found in (Dastjerdi and Buyya, 2016).

2.1.2 Trust

Trust is another crucial concept in our approach. Trust is defined as a personal, subjective “expectation a manager has about another's future behavior based on the history of their encounters” (Dastjerdi and Buyya, 2016), (dos Santos et al., 2016). The trust generally depends on the context in which the interaction between units takes place.

2.2 Distributed Access Control Architecture

The distributed access-control architecture combines the following components: RM, ACA, and the SLA established between fog nodes and the cloud service providers (CSPs).

2.2.1 Access Control Administrator (ACA)

a. Components of ACA

This module is used at each service layer of the CSP and fog nodes to ensure the access-control at the corresponding layer. Figure 3 shows the proposed model, which contains the following components:

- **Policy Enforcement Point (PEP):** This block is in charge of intercepting all the access requests of users. It ensures that the resources of the system can be accessed only if the policy authorizes it. In fact, the PEP extracts the authentication credentials, the user attributes and the context information from the access request.
- **Policy Decision Point (PDP):** It evaluates all necessary information/policies, according to the trust or the risk estimation, and the context. It requests behavior history of user from Monitoring System-Database (MS-DB). Then, it returns the grant or rejects decision to the PEP, which applies and enforces the decision.
- **Risk Evaluator:** Each access request and each permission is assigned a risk value within a particular context.

The proposed architecture provides a risk-based access control to enable a dynamic access to the available resources. The calculation of risk determines whether an event can cause a system to be destroyed and its assessment is useful for making decisions to avoid such damages.

- **Trust Evaluator:** The Trust evaluator uses the monitored information to identify the context, and calculate the trust value of each user.

If the trust value of a user decreases while a user has a session open, the RM sends a notification to the PDP, which re-evaluates whether the privileges that the user is exercising should be revoked. In this way, the system is able to deny access to misbehaving users before they can perform extensive damage to the system.

- **Monitoring System:** Each fog node and cloud server are supervised by a monitoring system. The monitoring is a process of continuous observation that reports any violation of the fog-CSP. Users have to achieve their assigned obligations and their final status reported by the monitoring system and forwarded to the MS-DB.

The final report stored in the MS-DB will be used to calculate the risk and the trust values. Besides, a monitoring function is also developed to check if the obligations are fulfilled or also to capture any unsuccessful decision-making, and then, deactivate such access permission when detecting suspicious activities.

- **Policy Information Point (PIP):** The policy of the system is stored in the PIP.
- **Trust DB:** The trust values are stored in the Trust Database.

b. ACA Procedure

Access requests of users, which include the requesting subject and the resource requested, are given to the PEP. The latter forwards the request to the PDP. The PDP takes the final decision concerning the access request by evaluating the risk that can engender the user and the trust about the user's behaviors. This is done by considering the attributes and the credentials of the requesting user. The PEP receives the decision from the PDP and then it either permits or denies the request. If it is allowed, it is forwarded to the RM of the SP for the deployment of requested resources.

The PDP retrieves the risk and the trust value of the user and determines whether the user is trustworthy or not. When the user is trusted enough to complete successfully, the access is granted and the Monitoring System will supervise it. In case the user is not trustworthy enough to fulfil one or more of the obligations that would be assigned to him, the system denies the access request.

2.2.2 Resource Manager (RM)

This module handles the resource needs of clients. This is a part of the access control mechanism of each CSP and fog nodes. Moreover, this module manages the resource requirements of each fog node and cloud service.

The RM is responsible for the deployment of resources. In fact, the requested resources are managed directly by this module if they are available locally. However, if the requested resources are stored in foreign services (remote data center), the RM contacts the ACA of the remote entity/node to provide the requested remote resources without repeating the access control mechanism. This procedure is taken owing to SLA established between the CSPs and fog nodes.

2.2.3 Resources Mapping

The access request received from the user contains the amount of required resources. After verification of the identity and making access decision, the PDP assesses if the services corresponds to a local fog node (the resources requested are locally hosted). If the service does not exist locally, it is assumed that this request needs services mapping by a relevant SLA. The ACA party invokes the SLA, if the requested resources are stored on a remote CSP.

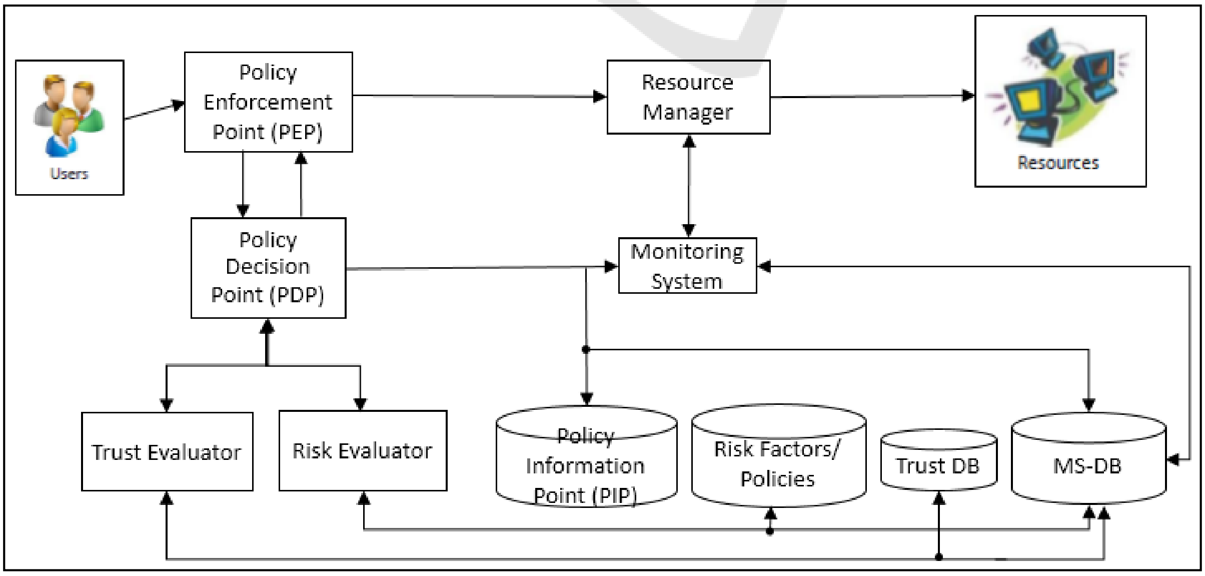


Figure 3: Overview of the access control administration (ACA).

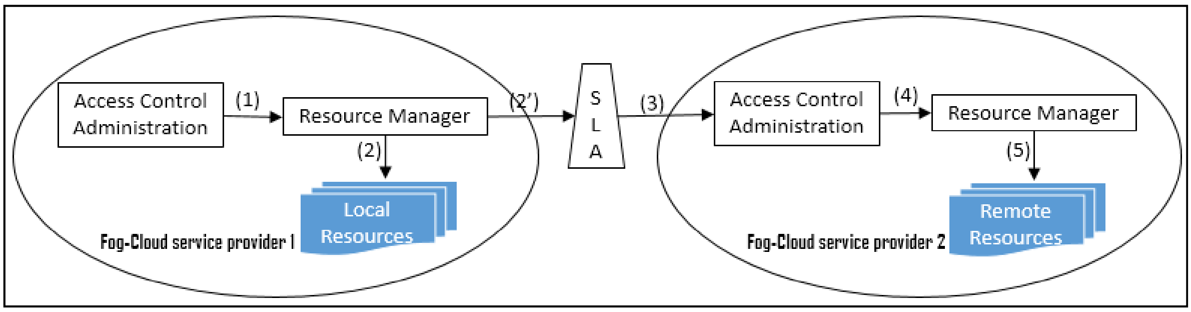

Access Control Administration
(1)
Resource Manager
(2)
Local Resources
Fog-Cloud service provider 1
(2')
S L A
(3)
Access Control Administration
(4)
Resource Manager
(5)
Remote Resources
Fog-Cloud service provider 2

Figure 4: Distributed authorization process in the multi_fog_cloud.

2.2.4 Service Level Agreement (SLA)

The SLA is a contract by which a CSP undertakes to provide a set of services to one or more customers (Almutairi et al., 2012).

In the fog-cloud environment, SLA ensures certain levels of security in the storage and management of personal data. It is then necessary to define very precisely different quality indicators that can be measured, analyzed and checked regularly. SLAs ensure access to remote resources while the resource manager is responsible for monitoring and checking deployed and demanded resources. To allow and facilitate exchange of information among the architecture entities, an SLA performs resource mapping. For this purpose and in order to prevent side-channel attacks, SLA implements separation of constraints for resource sharing. In the proposed approach, the ACA chooses whether to invoke the SLA or not depending on the location of requested resources (local or remote).

2.2.5 Distributed Authorization Process

As shown in Figure. 4, each fog node and CSP has both the ACA module and the RM module to deal with the access control and resource management in order to address the mobility of users.

When a user requests a service or local resource, the local ACA of the SP receives this request. If the ACA allows access to this user, it forwards the request to the local RM in order to have the requested resources. Otherwise, if the required resources are hosted in a remote cloud or remote fog node, the local RM contacts the ACA of the foreign fog/cloud as the appropriate SLA is already established between them.

After the receipt of the allowed access request, the ACA of the remote cloud or fog node forwards the request (since this request is permitted, the system does not repeat the procedure of the access control from the beginning) to the RM in order to assign the requested resources. Finally, the RM saves the identity of this user and monitors its behaviors, then stores these in the MS-DB.

3 IMPLEMENTATION AND RESULTS

In this paper, distributed access control architecture is proposed. In this experiment, we develop a script, which contains the activities process of our model. The script is implemented under Linux/Ubuntu. We deploy OpenStack to maintain the experiment. OpenStack is a software allowing the construction of private and public cloud. In addition, OpenStack is a community that is designed to help organizations implementing a server or virtual storage system.

For this work, several simulation experiments are provided for the server service provider as well as for the distant instance (Table 2).

Table 2: Real and Virtual Machine Condition of Simulation.

Condition	Openstack server	Remote instance
RAM	1 Go	2048 Mo
OS	14.04 LTS	12.04 LTS
CPU	2	1

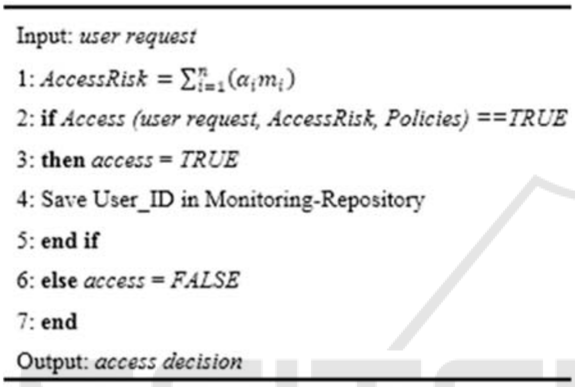
The experiment measures the estimated time to reach an access decision using the evaluation of trust and risk. When the user requests access to openstack/cloud, the script will be executed. Then, depending on the available resources, the system will calculate the risk and thus, it has an access decision. To calculate the risk value, we choose a limited number of risk factors to make our experience. These factors are evaluated for every access request and aggregated to achieve a measure of the total security risk. There are some factors that represent risks associated with the person or the application requesting access to a resource. Another ones are related to the risks associated with components in the path between requester and resource, such as the type of machines, and risks associated with the situation surrounding the request. Additionally, there are the heuristics, which represent the risk associated to previous similar requests, such as known violations. Therefore, equation (1) shown below presents the total risk:

$$\text{AccessRisk}= \sum_{i=1}^{n}(\alpha_i m_i) \quad (1)$$

where $\Sigma\alpha_i = 1$, α_i: is the weight, $m_{i:}$ is the used metric and n: is the number of metrics.

Algorithm 1 resumes the developed process.

Algorithm 1: Algorithm for the proposed access control scheme.



```
Input: user request
1: $AccessRisk = \sum_{i=1}^{n}(\alpha_i m_i)$
2: if Access (user request, AccessRisk, Policies) ==TRUE
3: then access = TRUE
4: Save User_ID in Monitoring-Repository
5: end if
6: else access = FALSE
7: end
Output: access decision
```

Our script contains risk calculation policies. At each access of such user, this script will be executed and we calculate the elapsed time to have a final decision about access. In this work, we choose 7 metrics to calculate the risk; each used metric has a weight of 1/7.

The risk factors used are:

- User_role: administrator, simple user …
- Previous violations: yes/ no.
- Machine type: Server, Pc, Mobile…
- Network: wireless/ wired
- Connection type
- Rank
- Distance between user and cloud data center.

Next, we compute the risk value based on the previous factors like in (2). These values in (2) are just an illustration and different systems can use different values.

$$\text{Risk\_metric1} = \begin{cases} 0.1 \text{ if user\_role} \in \text{Admin} \\ 0.5 \text{ if user\_role} \in \text{User} \\ 1.5 \text{ if user\_role} \in \text{otherwise} \end{cases} \quad (2)$$

We simulate our scheme several times successively and afterwards (7 runs), we averaged the results obtained from each simulation run.

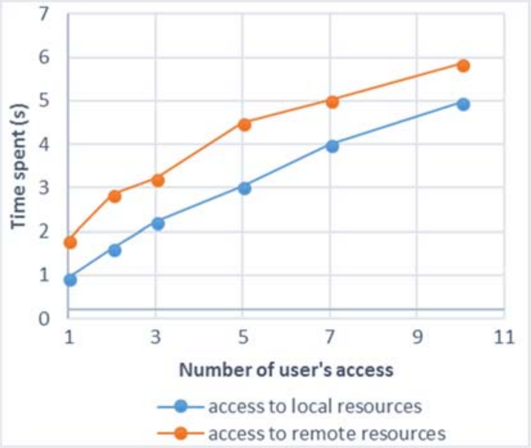


Figure 5: Average of time spent by number of user's access to local/ remote resources.

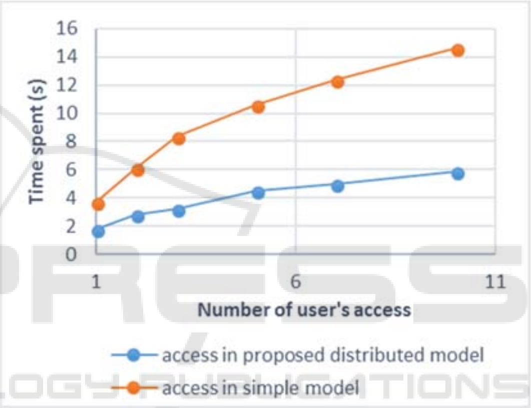


Figure 6: Comparison between distributed non-distributed models of access to remote resources.

From the results obtained in Figure. 5, we can see that our proposed scheme does not introduce an important time-based overhead. As our model is very distributed, when the number of users increases, the time spent to manage all requests remains short. Furthermore, by comparing the time spent to access local resources and the time spent to access remote cloud server, we can note that the differences are not very high since our access control model is distributed, which enable to function faster than others models.

Therefore, the delay elapsed to handle all access requests is not large. Owing to the fact that our model is associated with the monitoring function, when the user requests service and it is not available in this fog node, he/she will not repeat all the procedure by searching again in other neighboring fog nodes. By using our model in fog, the allocation of resources is realized with a short delay due to the distributed nature of fog. In fact, for higher number of dispersed services, the allocation results in a shorter delay.

Actually, if the user is trustable, he will have his requested services, whether this latter is either locally in fog nodes or far in foreign cloud data center. Hence, compared to other traditional solution like in (Dastjerdi and Buyya, 2016) the time spent for access to remote resources is decreasing. The strong point of our model is its ability to not overload the network and this by setting up an architecture of access control and monitoring taking place in a short time and indicating that it is fast and efficient. Furthermore, we can say that our model is secure. Thus, it is based on the calculation of risk and trust. Besides, the monitoring system is present to supervise and discover if there are violations or malicious activities. It intervenes to protect the system against destruction.

In the concept of the proposed solution, if the requested resources are located in foreign cloud services, the system is able to bring these services without repeating the access control procedure from the beginning. In fact, due to the monitoring system and the resource manager, the time spent will decrease towards half and that is a good result, which proves that the proposed scheme is effective. Moreover, the experimental results show higher performance when using a distributed architecture for controlling the access to the system.

4 CONCLUSIONS

The main characteristic of fog computing is its effective management of resources. The distributed nature of this environment and the deployment of an access control strategy bring many challenges such as deciding on the extension of collaborative work and the limit of resources sharing. Therefore, it is necessary to secure access to these resources. Hence, there is a need to develop a new distributed trust-based access control models that are adaptable to fog computing. In fact, to decrease heavy computational and communication overhead on service providers or data owners, we propose a dynamic and distributed access control strategy for fog computing based on risk and trust evaluation in order to improve the efficiency of fog resources' deployment and satisfy the users' security requirements.

We began by describing the fog paradigm and then identifying the need for a distributed strategy for controlling access to fog-cloud systems. Simulation was performed with an OpenStack platform in Linux. Simulation results show that our proposed distributed scheme can provide secure and efficient access control management for users and then improve the utilization of fog-cloud services.

REFERENCES

Almutairi, A., Sarfraz, M., Basalamah, S., 2012. A distributed access control architecture for cloud computing. *IEEE software, vol. 29, no 2, p. 36-44.*

Dos Santos D. R., Roberto M., Gustavo, R.S., Carla M. W., Carlos B. W., 2016. A Framework and Risk Assessment Approaches for Risk-based Access Control in the Cloud, *Journal of Network and Computer Applications*, vol. 74, p. 86-97.

Dastjerdi, A. V., Buyya, R., 2016. Fog computing: Helping the Internet of Things realize its potential. *Computer, vol. 49, no 8, p. 112-116.*

Huang, Q., Yang, Y., Wang, L., 2017. Secure Data Access Control With Ciphertext Update and Computation Outsourcing in Fog Computing for Internet of Things. *IEEE Access, vol. 5, p. 12941-12950.*

Hu P., Dhelim S., Ning, H., 2017. Survey on Fog Computing: Architecture, Key Technologies, Applications and Open Issues. *Journal of Network and Computer Applications, vol. 98, p. 27-42.*

Shirazi, S. N., Ul H., Gouglidis, A., Farshad, A., 2017. Review and Analysis of Mobile Edge Computing and Fog from a Security and Resilience Perspective. *IEEE Journal on Selected Areas in Communications, Issue: 99.*

Xiao, M., Zhou, J., Liu, X., 2017. A Hybrid Scheme for Fine-Grained Search and Access Authorization in Fog Computing Environment. *Sensors, vol. 17, no 6, p. 1423.*