A Security Framework for Fog Networks Based on Role-Based Access Control and Trust Models

Farhoud Hosseinpour, Ali Shuja Siddiqui, Juha Plosila, Hannu Tenhunen

A Security Framework for Fog Networks based on Role-Based Access Control and Trust Models

Farhoud Hosseinpour[1], Ali Shuja Siddiqui[2], Juha Plosila[1], and Hannu Tenhunen[1]

[1] Department of Future Technologies, University of Turku, Turku, Finland
Email: farhos; juplos; hatenhu@utu.fi
[2] Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, USA
Email: asiddiq6@uncc.edu

Abstract. Fog networks have been introduced as a new intermediate computational layer between the cloud layer and the consumer layer in a typical cloud computing model. The fog layer takes advantage of distributed computing through tiny smart devices and access points. To enhance the performance of the fog layer we propose utilization of unused computational resources of surrounding smart devices in the fog layer. However, this will raise security concerns. To tackle this problem, we propose in this paper a novel method using a trust model and Role Based Access Control System to manage dynamically joining mobile fog nodes in a fog computing system. In our approach, the new dynamic nodes are assigned non-critical computing tasks. Their trust level is then evaluated based on the satisfaction rate of assigned tasks which is obtained through different computing parameters. As the result of this evaluation, untrusted nodes are dropped by the fog system and nodes with a higher trust level are given a new role and privileges to access and process categorized data.

Keywords: Fog Computing, Cloud, Access Control, Trust model

1 Introduction

The benefits achievable by deploying scalable applications serving a large number of users simultaneously are rapidly generating novel innovations and expanding the reach of cloud computing. The cloud computing has replaced the need for owning large private data centers for service providers who want to deploy their projects with minimum infrastructure cost [1]. Cloud computing provides scalability for applications in manyfold by enabling addition and removal of processing nodes at runtime as needed. Although cloud computing has deemed itself useful in many scenarios [2], it is not viable for applications that require low latency and predictable feedback such as Smart Grids, industrial automation systems or intelligent transport systems [3]. This is due to the fact that systems in a cloud service are geographically distributed. For the alleviation of this issue, "*fog computing*" [4] has been introduced as a complementary concept to cloud computing.

Cloud computing can be defined using a layered computation model. Typically there are two layers: a cloud layer and a consumer layer. Recently a new computational layer, called a fog layer, has been introduced to the model. The fog layer resides between the cloud and consumer layers in the network's edge nodes like sensors and Internet-of-Things devices. Fog computing introduces the concept of location to cloud computing where traditionally non-locational computing has been dominant. Additionally, the fog layer also provides extra computational resources to the cloud layer.

Fog computing is currently an evolving new technology which aims to supplement already established cloud computing platforms to expand their application domain. Fog computing provides a location based expansion of the cloud by using heterogeneous computing devices and access points to which end nodes connect to communicate with the cloud. Bringing the computational intelligence geographically near to the end users provide new or better services for latency sensitive, location-aware and geo-distributed applications that due to their characteristics are not feasible merely through cloud computing. Delegating some simple yet frequent tasks of the cloud to the fog results in better performance for IoT-based applications [5]. In this paradigm, intelligent networking devices with both computation and storage capabilities, i.e., intelligent routers, bridges, and gateways, compose the fog computing platform near to the edge of the network. However, such devices are resource constrained and have computing and storage limitations.

Increasing computing capabilities of fog computing is a major challenge to improve the Quality of Service (QoS). To this end, one possible way is to leverage processing and storage capabilities of surrounding smart devices [6]. Smart devices have become an ubiquitous part of modern life. According to the Global Internet Phenomena Report Spotlight 2016 from Sandvine, the Waterloo-based broadband network equipment company in North America [7, 8]: " *The average household was found to have at least seven active, connected devices in use every day, while at the top end of the spectrum, 6 percent of households tuned in with more than 15 active devices, a marked increase over previous years. Whereas home roaming via mobile devices such as tablets and smartphones accounted for only nine percent of traffic five years ago, it now represents almost 30 per cent of home internet traffic across North America.*" Falaki et al. [9] developed a tool called SystemSens and investigated resource usage such as CPU, memory, and battery in smartphones. According to this study, except for the pick time between 11:00 to 17:00 the average CPU usage in all tested users is below 50%. This amount drops to less than 20% during the night time between 00:00 to 8:00.

Having this motivation, leveraging the available computing power of numerous different smart devices will enhance fog computing. However, utilizing resources of such devices for fog computing will impose some security challenges. In this paper, we present a novel approach to tackle this problem by leveraging containerization technology to provide isolation for fog computing tasks in external smart devices. This is further supported by role-based access control and trust models.

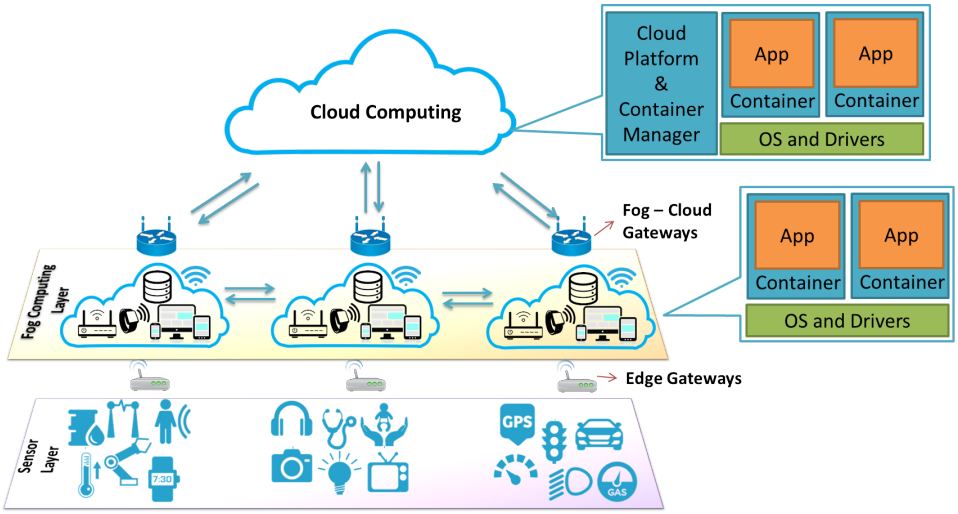

Fig. 1. Fog computing platform.

The rest of the paper is organized as follows. In Section II, we give an overview of the new emerging technology of fog computing. In Section III, related works to access control mechanisms for smart devices are presented and discussed. Then, in Section IV, we present our proposed framework. Moreover, a formal description of the proposed framework is presented in Section V, the results and discussion on the implemented model is presented in Section VI, and, finally, concluding remarks are given in Section VII.

2 Application of Fog Computing

Fog computing introduces an intermediate layer between the edge network or the end nodes and the cloud layer (Figure 1). The fog layer can be implemented using the same components as the cloud layer. The fog layer provides computation in a geographical location. It aims to provide a computing layer physically closer to the end node so that the computing capabilities can be brought near to consumers. The expected benefit is obtaining faster computation times for requests that require low latency. This can play an advantageous role in promotion of the Internet of Things (IoT) [10]. Utilizing fog computing reduces the overhead of communication with cloud trough internet and provides a faster response for applications which require lower latency. This is made possible by locally executing such processes in the fog layer and forwarding only those which do not require real-time computation or require higher processing power to the cloud layer. Schulz et al. [3] have investigated different latency critical IoT applications. According to their study, factory automation applications have the highest critical latency requirements in the range of 0.25 to 10 ms. Process au-

tomation, Smart Grids and intelligent transport systems are in the next place in their ranking.

In addition to the requirement of low latency, fog computing as middleware can pre-process raw data coming from the edge nodes before sending them to the cloud. Cloud computing, dealing with Big Data [11], has to process large amounts of data at any time. As a result, the fog layer not only reduces the amount of work needed in the cloud to generate meaningful results, but it can also reduce the monetary cost of computing in the cloud layer.

2.1 Fog Layer Structure

The most important and beneficial aspect of fog computing is the location proximity to the end nodes. The fog layer can be deployed on intelligent access points and gateways that not only connect the edge nodes to the cloud layer but also provide additional computing resources near the edge of the network. In addition to that, independent computing nodes such as smart devices can be added to the fog layer for the sole purpose of computation. Fog nodes can connect to each other to form a mesh. This can also be envisioned as a peer-to-peer (P2P) network with either centralized master controllers or a decentralized implementation without any controllers. Fog nodes cooperate and pool their resources to complete a task. The fog layer can be a dynamic network because some nodes might dynamically join and leave the network due to mobility or power limitations. Or, the other way around, the edge sensors might be mobile and move from one local fog network to another. A robust orchestration system is required to manage the execution of applications in such a dynamic environment without violating QoS and security.

Virtualisation: To support multi-tenancy of different applications and to achieve elasticity in large-scale shared resources, fog computing takes advantages of visualization technologies. A physical fog node can accommodate several virtual fog nodes. A fog computing platform is composed of several physical and virtual fog nodes that are deployed based on a hierarchical architecture [12] (Figure 2). Virtualisation technology based on Virtual Machines (VM) is not efficient or even feasible approach for resource constrained fog computing nodes. Containers are a new lightweight alternative for traditional VMs that are ideal for a fog computing platform. Containers provide OS level virtualisation without a need for deployment of a virtual OS. Hence, they are lightweight and significantly smaller in size than VMs. Containers provide a self-contained and isolated computing environment for applications and facilitate lightweight portability and interoperability for IoT applications [13]. Moreover, data and resource isolation in containers offers improved security for the applications running in fog nodes.

3 Related works

Fog computing was introduced in 2012 by Cisco [4] as an additional computing layer near to the edge of the network, to complement cloud computing services.

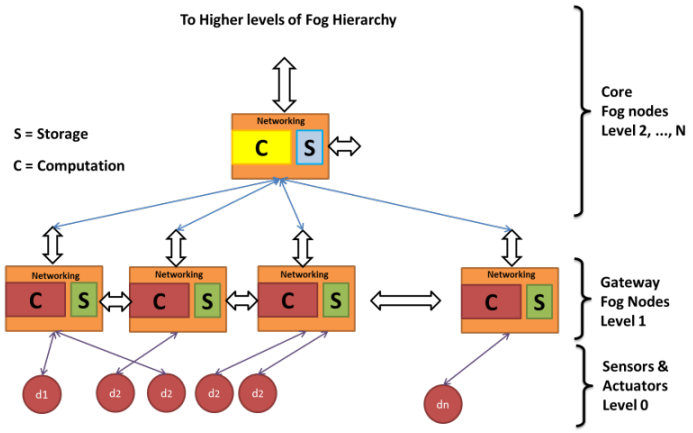

Fig. 2. Architecture of fog computing.

We discussed challenges for adoption of this technology in IoT applications as well as its security issues in our review paper [12]. Due to location proximity to the edge of the network, mobility of edge sensors, and also resource limitations, enabling scalable, flexible, and real-time strategies for resource allocation is very challenging. Oueis et al. [14] proposed a cluster-based resource allocation scheme for fog computing in which a cluster of fog computing resources is logically built depending on the profile of computation offloading request from an IoT device or a fog node. Yi et al. [15] investigated security and privacy issues of fog computing and pointed out that unlike the cloud computing that the cloud service provider owns all computing devices, fog computing is more flexible to leverage different computing resources belong to different parties. This flexibility adds more complexity in the terms of trust management and security. Misra et al. [16] proposed a cluster-based and multilevel hierarchical architecture for Wireless Sensor Networks (WSN) to establish an authentication mechanism. They deployed a multi-level access control system for each logical cluster using Role-Based Access Control (RBAC) model. They proposed a reputation-based trust model to assign a role for a node and form the logical cluster in WSN. They calculated the reputation value based on the behaviour of a node for successful transmission of data. Stavros et al. [17] addressed access control issues in fog computing for an intelligent transport system as a case study. They pointed out that fog computing has dispersed nature and sensors can enter and leave the network arbitrarily or the other way around, fog nodes could also be mobile. Hence, traditional identity-based authentication is not a feasible approach in this case. To cope with this problem, they proposed utilizing Attribute-Based Access Control (ABAC) model in which the authentication is based on the attributes of the subject (in this case, fog node) trying to access a data rather than their identity. In [18] the author discussed the importance of granting access based on the level of trust to individuals. They innovated a mobile device called MS-Ro-

BAC for implementation of role-based access control. The MS-Ro-BAC manages access and network authorizations through of role-based access control with no dedicated hubs, servers, special hard-drives or local administrators.

4 Proposed Framework

In this paper, we propose a framework for secure utilization of surrounding smart devices' processing capabilities in a fog computing platform. We use containers as virtualization technology in our fog platform. The reason for this is that they are: 1) lightweight and require less computing and storage, 2) easily portable, 3) platform independent and provide interoperability in a heterogeneous network and, 4) provide isolation of the application that utilize shared resources, which results in better security. We also design and develop an access control system based on the RBAC model to provide authentication for dynamic fog nodes joining the fog computing network. We consider three different kinds of fog nodes according to their capabilities and trust levels. As discussed earlier, a typical fog network is composed of smart communication nodes with the capability of acting as access points. This way they can communicate with edge sensors and also forward preprocessed data to an upper level in the cloud. Also, we propose utilization of dynamic nodes, each of which provides either processing resource only or combined processing and access point resource. Such nodes, after having been identified within the fog network, can join fog computing and share their resources.

Our framework employs trust models in transactions pertaining to data transfer and administration. This adds an extra layer of security and guarantees that untrusted nodes are not able to access sensitive data over the network. Dividing trust into levels will allow segregation of operations and data based on their criticality.

Whenever a node is made part of the fog for the first time, it is assigned the lowest level of trust and the least access privileges to the data to be processed as no knowledge of its previous transactions exists. However, after some transactions, the dynamic nodes can improve their reputation and gain a higher level of trust. They might also be disjoined from the network if any malicious actions are detected, or if they will no longer be in the vicinity of the computing environment. In cases like this, the nodes' access privileges need to be revoked. To make this possible, a manager node is required for managing the task allocation and participating nodes. Figure 3 illustrates the proposed framework in which the fog layer consists of four types of nodes: Fog Manager Node (FMN), Static Node (SN), Dynamic Node (DN) and Processing Node (PN). Any of these nodes have different roles and hence different privileges are assigned to them. A role for a node is defined based on its capability (Processing only or Processing and communication) and its current level of trust. The following section describes the definition of roles and trust in more detail.

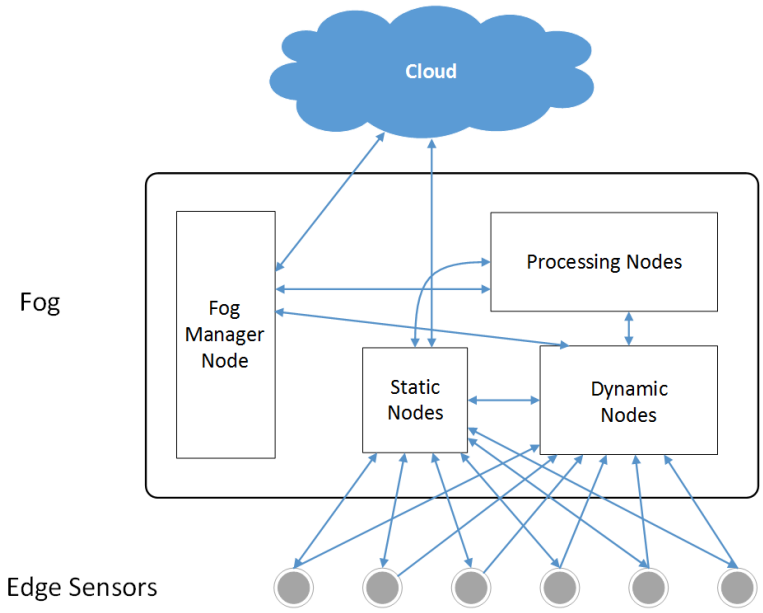

Fig. 3. Proposed Framework.

4.1 Roles

A role for a node defines its privileges for accessing different kinds of data and for participating in processing tasks. In this framework we assume three categories of information based on criticality: *non-critical*, *moderate* and *critical* data. The FMN assigns the roles to nodes based on their reputation and trust levels as well as their capabilities. We define four roles according to the node types in this framework. With each role within fog layer, a set of permissions is assigned. This set limits the nodes' access to certain types of data and defines their privileges and responsibilities in processing certain tasks. Table 1 summarize the security privileges of each node according to its role.

A detailed description of each node is presented in the following.

Fog Network Manager is the central overseer of the fog network. Whenever a node wants to join the network, it must contact the FNM. If the connecting node is an edge node, the FNM will send this node the address of the active fog nodes based on location proximity. The edge node will then connect to the nearest fog node and start sending its data. In case a fog node goes offline, the edge node will be provided with the address of the next suitable nodes to connect to. If a smart device attempts to join the fog network, the FNM will assign the connecting node with the lowest trust level. On the other hand, if a node needs to be deleted from network (due to malfunctioning or permanently disconnecting), the FNM will revoke access rights from that node and update

Table 1. Assignment of privileges according to roles.

	Privileges						
Role	Processing	Edge communication	Cloud communication	Verifying the tasks	Assigning a task	Adding new nodes	Revoking access
FNM		×	×	×	×	×	×
SN	×	×	×	×			
DN	×	×					
PN	×						

the list of active fog nodes to the edge sensors as well as the cloud layer. The FNM is also responsible for promoting or demoting the roles of participating fog nodes according to their trust levels.

Static Nodes These nodes are used for connecting edge devices to the fog layer. They are static in nature and are expected to be available at all times. By default, they are assigned the trust level *High*. They can either process data themselves or they can request the FNM to initiate a task on some other processing node or an access point. Upon completion of a task, they can forward the data themselves to the cloud layer.

Dynamic Nodes are dynamic and are intended to be used as access points as well as processing nodes. They start with trust level *Low* and gain more levels as trusted more by the FNM. They do not themselves send data to the cloud layer but, instead, they use the static access points for the purpose.

Processing Nodes are dynamic and are used exclusively for processing data. Since they are dynamic, their initial trust level is *Low* and it is increased as the node gains more trust. They cannot themselves connect to the cloud layer nor act as access points but, instead, they use static access points for sending their data. These nodes only share their processing resources. Therefore, their tasks are assigned by the network manager or static nodes. After processing the task the results needed to be sent to a static nodes to be forwarded to the cloud layer.

4.2 Trust Management

A trust level is a measure of the reliability of a participating node. Trust management is applied to dynamic fog nodes of the network. Static fog nodes at any time are considered to have the highest level of trust. Trust in our system is defined in terms of a nodes' privilege to process a certain type of data. It can be divided into multiple levels but in this paper, we will consider the division into three levels:

- *Low*: This is the lowest level. Dynamic fog nodes are initially assigned this level upon joining the network. The FMN assigns tasks of the lowest priority and criticality to these nodes. Data computed by nodes with this trust level is sent to one of the static nodes to be verified before sending to the cloud layer.
- *Moderate*: This is the second level of trust. On this level, the data is considered to be of moderate criticality. The fog node handling this data is assumed to be reliable, and the result generated by the node will be sent directly to the cloud layer. Dissatisfaction in the service of nodes in this level will demote the node to the low level. However, dissatisfaction up to a pre-determined level can still be tolerated.
- *High*: This is the highest level of trust. The data which is considered to be most critical by the application is handled by the nodes at this level. The requirement of processing is not only that the data be processed correctly, but also that the nodes maintain the highest level of service. The data processed by nodes in this level is sent directly to the cloud layer.

The trust level of each dynamic fog node evolves over time and can change on interaction with other nodes. We utilize already established trust algorithms for our implementation. There can be several ways to calculate the trust level for a node. Manuel et al. [19] investigated different factors to evaluate trust value of a recourse in cloud computing. They claim that combination of multiple trust factors such as availability, reliability, data integrity, and turnaround efficiency should contribute to the trust model of a resource. According to this study, in our proposed framework we calculate the trust value based on all attributes mentioned above.

In the following we discuss each of these attributes and present a formula to compute the trust value of a resource based on those attributes:

Availability is a measure to ensure that a resource is operational and accessible to authorized parties whenever needed. A resource is deemed unavailable if 1) it is too busy to process and responds a task request, 2)it denies a task request, or 3) it is just shut down. Availability of a resource Av_R is calculated based on the following equation over a period of time:

$$Av_R = \frac{Ac}{Sb} \tag{1}$$

where Ac denote the number of computing tasks accepted by a resource and Sb denote the total number of tasks submitted to that resource.

Reliability or success rate of a resource is a measure and quality of a resource in consistently performing according to its specifications in specified time. Reliability of a resource Re_R defines its success rate in the completion of the tasks that it has accepted and is calculated based on the following equation over a period of time:

$$Re_R = \frac{Cs}{Ac} \tag{2}$$

where Cs denote the number of accepted tasks completed successfuly by a resource, and Ac is the total number of accepted tasks by that resource.

Data Integrity involves maintaining the consistency, accuracy, and trustworthiness of data over its entire lifecycle. Integrity ensures that information is not modified by unauthorized entities. Data Integrity of a resource Di_R is calculated based on the following equation over a period of time:

$$Di_R = \frac{Cm}{Ac} \tag{3}$$

where Cm denote the number of tasks that a resource successfully preserves data integrity, and T is the total number of accepted tasks completed successfully by a resource.

Turnaround Efficiency is a quality that a resource accomplishes a task within the time that it promises. Turnaround is a time frame that starts from when a broker sends a processing request to a resource till the time that the resource completes the task successfully. Turnaround Efficiency of a resource Te_R is calculated based on the following equation over a period of time:

$$Te_R = \frac{Pt}{At} \tag{4}$$

where Pt denote the Promised Turnaround time by a resource for completion of a task and At is the Actual Turnaround time by a resource for the completion of a task.

Trust Value of a resource: The overall trust value for a resource is calculated based on composition of all attributes of a resource with following equation:

$$TrustValue_R = (a * Av) + (b * Re) + (c * Di) + (d * Te) \tag{5}$$

where $a+b+c+d = 1$ are coefficient positive numbers that define the weight of each attribute and Av, Re, Di, and Te are respectively average value for Availability, Reliability, Data Integrity, and Turnaround Efficiency over determined time T.

Task assignment done by the FNM is also dependent on the trust levels. To ensure that each trust level would have the required number of nodes to perform all tasks defined for that trust level, we will use the weight function to evaluate the need to increase the trust level of a dynamic node. It would be preferred for a node to be promoted to a higher trust level if there is a shortage of nodes at a higher level. For each dynamic fog node, the weight is calculated as:

$$Weight = \begin{cases} 1 - \frac{N_{req}}{N_{avail}} \; if & N_{req} < N_{avail} \\ 0 & otherwise \end{cases} \tag{6}$$

where N_{req} is the number of nodes required at the next higher trust level, and N_{avail} is the number of nodes available at the next higher trust level.

5 Results and Discussion

The fog computing platform is implemented in SystemC environment. Each processing unit is modeled by a SystemC module which can communicate with all the other processing elements in its domain through a SystemC channel. We have considered heterogeneous nodes with different processing capabilities. The execution frequency for each processing elements varies between 500MHz up to 4GHz. Applications enter and leave the system based on a randomized amount of workload during the time. Each application is modeled as a task graph where each task should be assigned to a processing element exclusively. Execution of the tasks are independent of each other, and only the data transfer between tasks connects two tasks to each other. Therefore each task can be run at a different frequency. The fog system comprises of a number of fog nodes which include static nodes, processing nodes, and a fog manager node. Along the time, a group of dynamic nodes joins and leave the fog system. The fog manager assigns the tasks to the newly joined dynamic node and calculates their trust level based on the Trust formula. So, once any of the new dynamic nodes reaches to the desired trust level, then the fog system upgrades their role to become trusted fog node.

Figure 4 (a) shows the number of completed applications in the fog system during the time. Figure 4 (b) illustrates the total number of the nodes once the dynamic nodes join and leave the fog system. The dashed line shows that the system is able to detect and eliminate the untrusted nodes in each interval. As it can be seen, while the number of nodes in the fog system increases, the rate of the completed applications also increases. And finally 4 (c) shows the total number of identified untrusted nodes during the execution time.

6 Conclusion

The fog computing paradigm extends cloud computing and services to the edge of the network to support geographical distribution and mobility of end users. In this paper, we presented a security framework for fog computing infrastructure. After discussing the potential application of fog computing, we argued that to increase the performance of the fog layer we can take advantages of vacant resources of surrounding smart devices such as smart phones and tablets. To tackle the security issues that are imposed by this technique we proposed an implementation of role-based access control in conjunction with trust models. We presented how our proposed framework can contribute to solving security issues of a fog network. In our framework, we defined a method to calculate trust

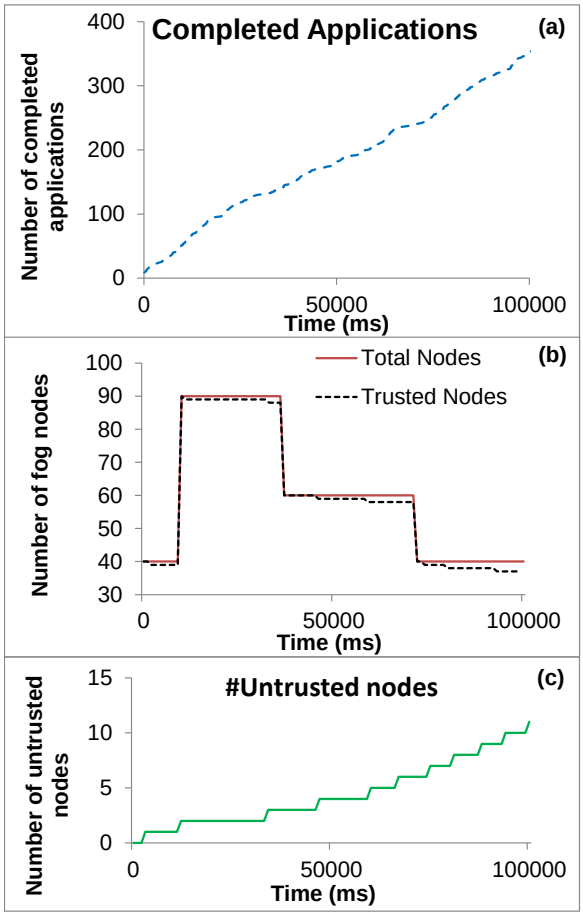

Fig. 4. Experimental results.

levels based on computing tasks assigned to the nodes. Moreover, we presented algorithms for implementation of our framework. According to our implementation results, the fog system was able to distinguish the trusted and untrusted dynamic nodes. However, in addition to secure access control and authentication methods, secure computation schemes need to be undertaken to guarantee the security and integrity of data in a fog network.

Acknowledgment

This work was supported by University of Turku Foundation, EIT Digital and the Department of Information Technology - University of Turku.

References

1. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.
2. B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *2009 Fifth International Joint Conference on INC, IMS and IDC*. IEEE, 2009, pp. 44–51.
3. P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel, A. Puschmann, A. Mitschele-Thiel, M. Muller, T. Elste, and M. Windisch, "Latency critical iot applications in 5g: Perspective on the design of radio interface and network architecture," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, February 2017.
4. 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*, ser. MCC '12. New York, NY, USA: ACM, 2012, pp. 13–16.
5. V. K. Sehgal, A. Patrick, A. Soni, and L. Rajput, *Smart Human Security Framework Using Internet of Things, Cloud and Fog Computing*. Cham: Springer International Publishing, 2015, pp. 251–263.
6. W. Shi, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637 – 646, 2016.
7. S. I. B. Networks, "2016 global internet phenomena report, latin america & north america," Sandvine - Intelligent Broadband Networks, Tech. Rep., 2016.
8. J. MacLean, "Households now use an average of seven connected devices every day," Cantech Letter, Tech. Rep., 2016.
9. H. Falaki, R. Mahajan, and D. Estrin, "Systemsens: A tool for monitoring usage in smartphone research deployments," in *Proceedings of the Sixth International Workshop on MobiArch*, ser. MobiArch '11. New York, NY, USA: ACM, 2011, pp. 25–30.
10. K. Ashton, "That 'internet of things' thing," *RFiD Journal*, 2009.
11. J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Byers, *Big data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute, 2011.
12. F. Hosseinpour, Y. Meng, T. Westerlund, J. Plosila, R. Liu, and H. Tenhunen, "A review on fog computing systems," *International Journal of Advancements in Computing Technology(IJACT)*, vol. 8, no. 5, pp. 48–61, 2016.
13. P. Bellavista and A. Zanni, "Feasibility of fog computing deployment based on docker containerization over raspberrypi," in *Proceedings of the 18th International Conference on Distributed Computing and Networking*, ser. ICDCN '17. New York, NY, USA: ACM, 2017, pp. 16:1–16:10.
14. J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015, pp. 1–6.
15. S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *The 10th International Conference on Wireless Algorithms, Systems, and Applications (WASA)*, 2015.
16. S. Misra and A. Vaish, "Reputation-based role assignment for role-based access control in wireless sensor networks," *Comput. Commun.*, vol. 34, no. 3, pp. 281–294, Mar. 2011.

17. S. Salonikias, I. Mavridis, and D. Gritzalis, *Access Control Issues in Utilizing Fog Computing for Transport Infrastructure*. Cham: Springer International Publishing, 2016, pp. 15–26.
18. T. House, "Mobile secure role based access control (ms-ro-bac) device," in *SoutheastCon, 2005. Proceedings. IEEE*, April 2005, pp. 542–546.
19. P. Manuel, "A trust model of cloud computing based on quality of service," *Annals of Operations Research*, vol. 233, no. 1, pp. 281–292, 2015.