Evaluation of Attribute-Based Access Control (ABAC) for EHR in Fog Computing Environment

Aisha Mohammed Alshiky, Seyed M. Buhari, Ahmed Barnawi
IT Department. Faculty of Computing and Information Technology
King Abdul-Aziz University
Jeddah, Saudi Arabia
e-mail:aalshikey@yahoo.com , mesbukary@kau.edu.sa , ahmed_barnawi@hotmail.com

***Abstract*—**Fog computing - a connection of billions of devices nearest to the network edge- was recently proposed to support latency-sensitive and real time applications. Electronic Medical Record (EMR) systems are latency-sensitive in nature therefore fog computing considered as appropriate choice for it. This paper proposes a fog environment for E-health system that contains highly confidential information of patients Electronic Health Records (EHR). The proposed E-health system has two main goals: (1) Manage and share EHRs between multiple fog nodes and the cloud,(2) Secure access into EHR on Fog computing without effecting the performance of fog nodes. This system will serve different users based on their attributes and thus providing Attribute Based Access Control ABAC into the EHR in fog to prevent unauthorized access. We focus on reducing the storing and processes in fog nodes to support low capabilities of storage and computing of fog nodes and improve its performance. There are three major contributions in this paper first; a simulator of an E-health system is implemented using both iFogSim and our iFogSimEhealthSystem simulator. Second, the ABAC was applied at the fog to secure the access to patients EHR. Third, the performance of the proposed securing access in E-health system in fog computing was evaluated. The results showed that the performance of fog computing in the secure E-health system is higher than the performance of cloud computing.

Keywords- *Fog computing, Electronic Medical Record (EMR), Electronic Health Record (EHR)*

*****

I INTRODUCTION

The explosive increase in the use of sensors and sensing information leads to the scope of producing plenty of future applications. The most important requirement in these applications is low-latency processing and as known centralizing of services may lead to high latency which is rejected in these applications. Although there are numerous economic advantages of cloud, there is a problem for latency-sensitive applications due to frequent movements of huge data from the source to the server/cloud [1].

The latency-sensitive and real time applications require nodes in the vicinity to provide fast responses. A new platform is needed to achieve these requirements; [2] Cisco recently proposed a new computing environment called fog computing, call it “Fog”, simply because fog is a cloud close to the ground. It is a connection of billions of devices (called as fog nodes) around the globe. Cloud computing defers from fog computing in the distribution of processing in distributed nodes with mobility. In fog computing environment, the generic application runs logic on resources throughout the network, including dedicated computing nodes and routers [3]. "The emerging fog computing architecture is a highly virtualized platform that provides compute, storage, and networking services between end devices and traditional cloud computing data centers, typically, but not exclusively located at the edge of the network" [4].

However, developing applications using fog computing resources is critical because it includes heterogeneous resources at different levels of network hierarchy to provide low latency and scalability requirement for new applications [3].

We consider the fog environment as an appropriate platform to deploy and support the Electronic Health Records (EHR). Nowadays, in modern healthcare environments, healthcare providers are shifting their electronic medical record systems to clouds [5]. Knowing that the cloud is not a good choice for real time and latency sensitive applications, we propose that the fog computing is appropriate choice for E-health system and to support EHR real time environment. EHR contains private and sensitive patient health information which is needed to be secured and the privacy of the patient must be ensured. Therefore, security in fog computing environment will eventually become an issue; with security embedded into the fog computing environment, we envision, in this research, to provide appropriate security solutions without affecting the performance level. With the proposed Attribute Based Access Control (ABAC) which is a flexible and logical mechanism [6], we will serve different users based on their attributes, object (information and resources) attributes and environment conditions (time and location). Thus, providing secure access mechanism into the EHR fog to prevent unauthorized access to fog and also prevent leaks of information; user-based attributes might be related to a targeted application such as if the target application was health the user attributes in this application will be for example user role (nurse/ doctor or so on), user job (specialist/ consultant or so on) and any attributes related to this health application will be present user attributes.

Therefore, in this research, there are three major contributions first, we simulate E-health system by using iFogSim tool and we come up with iFogSimEhealthSystem simulation tool. Second, we applied ABAC at fog to secure access into EHR of patient. Third, we studied the performance of proposed securing access in E-health system in fog computing.

This paper document of five chapters other than this one. Next chapter will give the literature review that discusses security of cloud for healthcare system and some of studies that implement healthcare system in fog will present in third chapter. The fourth chapter will present the framework design and implementation of E-health system including system architecture and main classes of proposed simulation tool.

The implementation will be evaluated in the fifth chapter to check the performance and efficiency of our proposed ABAC which applied on E-health system in fog computing. Finally the sixth chapter concludes this document and states the future work.

II LITERATURE REVIEW

Instead of cannibalizing Cloud Computing, Fog Computing allows a new type of applications and services, and that there is a rich interplay between the Cloud and the Fog, mainly when it comes to data management and analytics. This review is mostly related to work and deals with the potential risks of privacy exposure to the healthcare system and implement electronic health record (EHR) in fog computing [1]. Security in Fog Computing Environment will eventually become an issue; this issue is not being investigated yet and it seems to be completely absent in the literature. For that, this section discusses a number of related and similar researches that provide security of cloud system especially for EHR.

One of studies [7] explains that patients' records must be accessible only by authorized users and they justified that patients should have the opportunity to exert the control over their own data. For that, they proposed a cryptographic access control scheme allowing patients to grant medical teams authorizations to access their medical data. They proposed a schema consists of decentralized hierarchical key agreement protocol to securely establish a hierarchy of crypto keys in agreement with the privilege levels of the team members. The scheme provides data confidentiality, but it must be guaranteed that hierarchical keys are unique and "fresh" for each run of the protocol which require high computation.

As multiple entities will interact with the data, the authors in [8] explain that access to sensitive resources should be provided only to authorized users and tenants. They adapt Task-Role-Based Access Control, which considers the task in hand and the role of the user. They support both workflow based and non-workflow based tasks and authorize subjects to access necessary objects only during the execution of the task. Classification of tasks and activities has been done on the basis of active and passive access control and inheritable and non-inheritable tasks. Each user is assigned a role, roles are assigned to workflow or non-workflow tasks, and tasks are assigned to permissions. This model only supports the scenarios when the roles are defined within a single healthcare organization. It is designed to support healthcare service provided in a single healthcare organization. So, the access should be restricted and provided only during the execution of a specific task.

In [5] and [9], the authors mainly focuses on access control issues when EHRs are shared with various health care providers in cloud computing environments. In [5], they proposed a unified access control scheme which supports patient-centric selective sharing of virtual composite EHRs using different levels of granularity, accommodating data combination and various privacy defense requirements. However, this approach assumes that all health care providers adopt a unified EHR schema, which is not applicable in cloud environments. In [9], the authors try to overcome this limitation by supporting EHRs aggregation from various health care providers considering different EHR data schemas in cloud environments. They propose a systematic access control mechanism to support selective sharing of composite electronic health records aggregated from various health care providers in the cloud. They present algorithms for EHRs data schema composition and cross-domain EHR aggregation.

In [10], the authors explain that Attribute-Based Encryption ABE (data can only be read by a user with certain attributes [10] suitable for electronic health records system in the cloud, in which many users can retrieve the same EHR while each user can only decrypt the parts that they are allowed to read. The authors here try to handle some problems such as when a user with multiple roles might cause information leakage and computational overhead on EHR owners. Hence, they adopt both ABE and Identity Based Encryption IBE (a type of public-key encryption in which the public key of a user is unique user identity) and integrate them into their hierarchical framework. ABE is used to achieve fine-grained access control while IBE is used to securely transmit ABE keys. EHRs are encrypted on the Trusted Server and then are uploaded to the cloud. Decryption keys are also generated on trusted server and are distributed to domain servers that are then responsible for distributing the decryption keys to authorized entities. This framework addresses only the case of read access. This solution was suitable for an environment which has large number of users (subject) because it depends on their attributes which need not be predefined for each user.

Many research works proposed important and useful concepts of the EHR security [5, 7, 8, 9, and 10]. However, there are several uncertain issues. One of those issues is how to manage information of PHR and bring it near the user to support quick access of these information in timely manner. Therefore, allowing a hospital staff to access patient information (EHRs) in short period is essential. Information stored in the patient’s EHR may help a medical staff to make better decisions. In some emergency healthcare situations, immediate exchange of patient's EHRs is crucial to save lives. In our research, we try to handle the EHR near to the medical staff and provide quick response for patient needs. We will support that by implementing part of EHR in suitable and nearest fog nodes and we propose that Attribute Based Access Control (ABAC) that depends on attributes of subject (who want to access), object (services or information), action attributes (view or delete patient information) and environment conditions (time and location). This approach is flexible and it decreases the administrative overhead [6].

III FOG COMPUTING APPLICATION IN HEALTHCARE

In this section, we will review some of studies that applied fog computing in health care system. How to develop real-world fog computing-based universal health monitoring system is still an open question.

In [11], pervasive fall detection is employed for stroke mitigation. There were four major contributions in this study: (1) they examined and developed a set of new fall detection algorithms built on acceleration magnitude values and non-linear time series analysis techniques, (2) they designed and employed a real–time fall detection system employing fog computing paradigm, which distributes the analytics through the network by splitting the detection tasks between the edge nodes (e.g., smartphones attached to the user) and the server (e.g., cloud), (3) they examine the special needs and constraints of stroke patients and they proposed patient centered design that is minimal intrusive to patients and (4) their experiments with real-world data displayed that their proposed system achieves the high specificity (low false alarm rate) while it also achieves high sensitivity. Depend on researchers knowledge, their proposed system is the first large scale, real-world pervasive health monitoring system that employs the fog computing paradigm and distributed analytics.

Ultraviolet (UV) radiation has a great effect on human health. Since sensors in mobile phone cameras are very sensitive to UV, mobile phones have the potential to be an ideal equipment to measure UV radiance. The research [12] investigated theoretical foundations that control mobile phone cameras without any add-on to measure solar UV in open environment. Theoretical foundations accomplished to a procedure that can be deployed to any mobile phone with a camera. In addition, by utilizing fog computing, results can be collected and edited locally through fog server to provide accurate UV measurement. Furthermore, an Android app called UV Meter was established based on the procedure that can be implemented in mobile phones. Verification was conducted under unlike weather conditions and their results showed that the procedure is valid and can be implemented onto mobile phones for everyday UV measurement.

In another study [13], efficient IoT-enabled healthcare system architecture which benefits from the concept of fog computing is presented. The effectiveness of fog computing in IoT-based healthcare systems in terms of bandwidth utilization and emergency notification is demonstrated. In addition, they utilized ECG feature extraction at the edge of the network in their implementation as a case study. They proposed that to perform functionalities of gateways, the smart gateway should have the ability to offer a high level of advanced services in the fog computing platform. The smart gateway architecture including physical and operational structures is elaborately designed and described.

IV FRAMEWORK DESIGN AND IMPLEMENTATION

A. *System Architecture*

In our proposed E-health system architecture design, there are three fog nodes (each fog in a separate network) and one cloud. The fog nodes in our E-health system is used to serve three networks (reception, laboratory and clinic1) while, the cloud is used to serve the entire hospital. And there are three type of network communication:

- Device-to-Cloud communication. We assumed that there are administrators who are responsible for the administrative tasks of controlling user access and he/she connects to fog/cloud from PC or laptop to do their jobs.
- Device-to-Fog communication, for example, the reception employees use their device to look for patient information during the connection to reception fog in this network.

- Fog-to-Cloud Communication, for example, when the cloud receives an EHR request from reception fog when the patient visits hospital.

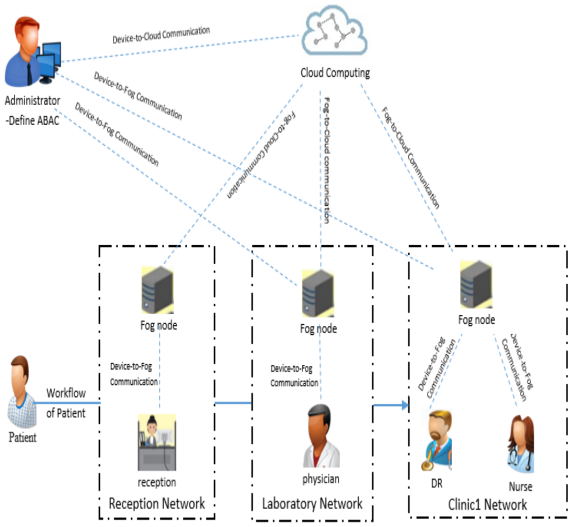
Device-to-Cloud Communication
Cloud Computing
Administrator
-Define ABAC
Device-to-Fog Communication
Device-to-Fog Communication
Fog-to-Cloud Communication
Fog-to-Cloud communication
Fog-to-Cloud Communication
Fog node
Fog node
Fog node
Device-to-Fog Communication
Device-to-Fog Communication
Device-to-Fog Communication
Device-to-Fog Communication
Workflow of Patient
Patient
reception
Reception Network
physician
Laboratory Network
DR
Nurse
Clinic1 Network

Figure IV.1. System network architecture

Figure IV.1 shows the general architecture of our proposed design and assumed patient workflow. We proposed simple unified workflow for each patient who visits hospital. This workflow helps us to explain the managing and sharing of patient EHRs between fog nodes and cloud. Also, it helps to explain how and where access policy will be defined and applied.

Figure IV.2 explain an example of the arbitrary assumption of patient movement in hospital and timeline of this movement. We assumed that there is specific timeline from the time the patient arrives to the hospital until he/she leaves. This timeline determines the minimum expected time of patient arrival (TMinA) and the maximum expected time of patient arrival (TMaxA) to (Reception/ Laboratory/ Clinic1) in unit of seconds. Also, it determines the maximum allowed time that is needed at each department for patient service in unit of minutes (TMaxPS).

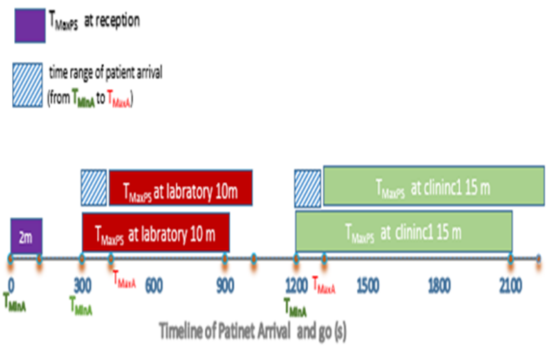

Figure IV.2. Assumed timline of patient arrival and go (s)

On the other hand, the communications and operations between network devices, fog and cloud (data center) can be summarized as follow:

- Patient visits hospital and go to reception.
- The reception employee look for EHR of patient by entering patient number (PID). This request of EHR will be sent to reception fog where it will be redirected to cloud. After the cloud receives the request of EHR, it checks the predefined access policies (ABAC). Depending on the access policies the access is permit if his\her access allowed then the rows of EHR will be submit to reception fog with timer (time of EHR availability in this fog). Reception fog stores patient information (rows of EHR) and starts a timer. Then, reception fog resends this patient information to end user.
- After sending the EHR to reception fog, the cloud will send same patient EHR to specific fog depend on estimated patient workflow. Usually EHR of patient will be available at laboratory or clinic1 fog before patient arrives. The availability of EHR near the end user instead of cloud will support high response time and low latency.

Note: as said above we take in consideration that low capabilities of fog storage and its computing and we proposed temporary storing of EHR in fog. We assumed that there is a specified timer with specfic EHR when it comes from the cloud. If this timer ends, then EHR of specific patient will be deleted from fog storage.

- T_{store} is the time when fog store received broadcasting EHR from cloud.
- $T_{avalliabilty}$ is timer of EHR availability in fog which is specified by cloud
- T_{delete} is the Time to delete the EHR from fog

$$T_{delete} = T_{store} + T_{avalliabilty} \qquad (1)$$

B. E-health Application Classes Design

For implementing functionalities of iFogSimEhealthSystem architecture, we leveraged basic event simulation functionalities found in iFogSim [14]. Entities in iFogSim, like FogDevice, communicate between each other by message passing operations.

We created some of classes that we need in our implementation and we edit some of existing classes to face needs of our application such as FogDevice, Tuple, Controller and ModulePlacementEdgewards. The most important classes which we generate are:

- **EndUser:** act as end user device of network described in the architecture. In this class the tuples (request of EHR) are generated and transmitted to FogDevice class and it is responsible for computing total response time for each request.
- **PolicyAuthorization:** is responsible for checking the authorization of EHR request and send authorized request to PolicyEnforcer class.
- **PolicyEnforcer:** is responsible for enforcing access policy. This class contact with two classes to complete its tasks PolicyAdministarator and AttrbuteProvider.
- **PolicyAdministarator:** manages and evaluates predefined policy statement and add a new policy statement.
- **AttributeProvider:** manages and retrieves missing attributes which are needed to enforce access policy.
- **PolicyStatment:** class stores information of policy statements.
- **User:** class stores information hospital department users with their IDs. We assumed that the users are reception, physician, specialist, consultant and urse.
- **Patients:** class stores information (EHRs) of patient at hospital. We assumed that patient EHR consist of general profile information, test lab required and diagnoses.
- **HealthSystem:** in our simulation the main class is HealthSystem that is responsible for creating the proposed network topology and application.

We designed the modeling of application at our system by using application (programming) models: The applications developed for deployment in the fog are based on the Distributed Data Flow (DDF) model. The application is modeled the same as a collection of modules, which represent the data processing elements. This application model allows us to represent an application in the form of a directed graph, with the vertices representing application modules and directed edges display the flow of data among modules. The information mined from the incoming streams is stored in data centers for large-scale and long-term analytics. Figure IV.3 shows application model of EHR request coming from reception the circle represent module while the line between modules represents edges and communication between modules. While **Error! Reference source not found.** shows application model of EHR request coming from laboratory or clinic1 end user. In our E-health application we created a class for each module as it will be shown in this section.

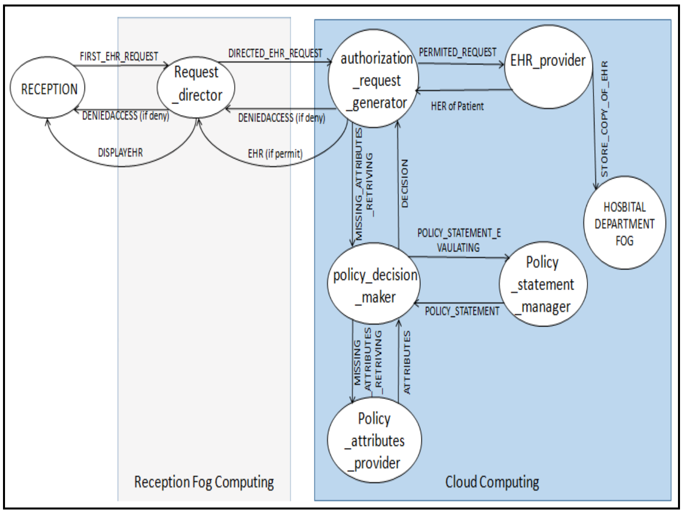

Figure IV.3. DDF of reception request

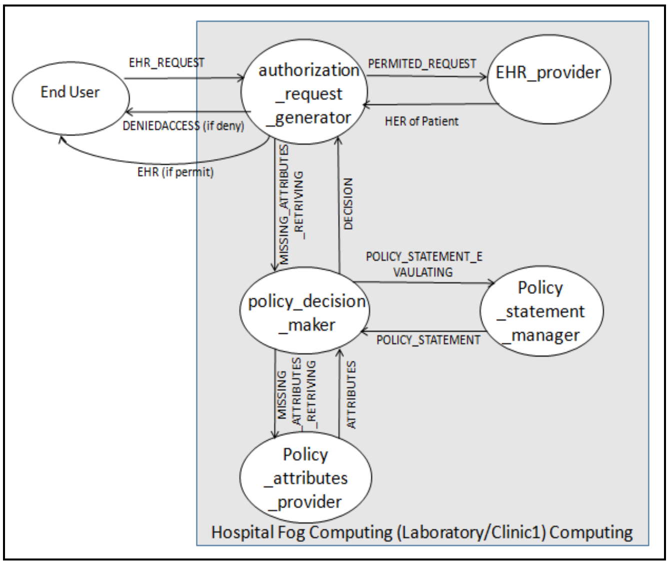

Figure 4. DDF of Laboratory or Clinic1 user request

V PERFORMANCE EVALUATION

In this section the evaluation of the performance of the proposed framework is presented. The objectives of these tests are to examine whether the using of ABAC framework will secure and control the access of users into the E-health system; and its efficiency in saving the time.

The test is conducted in two different environments, the first is cloud computing where there are several clients that request their services from one cloud, while the second environment is fog computing, where there are a number of simulated fog servers near to the clients and each fog server serves specific clients which are existing in its network boundaries. In addition, the test is repeated with different test inputs to confidently evaluate the system validation and performance. In each run of the test, the following information is collected.

- Request start time: time at which the client transmit tuple
- Request end time: the time at which the client recived tuple arrival
- Execution start time: the time at which the tuple actually starts at fog/cloud.
- Execution end time: the time at which the tuple finished at fog/cloud.

These pieces of information are used in calculating the following performance metrics.

- Execution Time (EXT_t) for specific sending tuple:

$$EXT_t = \sum mEXT_t + mTT_t \qquad (2)$$

Where $mEXT_t$ is the estimated execution time at a specific module and mTT is the transmission time between modules. As mentioned in the workflow design section, the tuple of EHR request have multiple modules to be executed and each module have a specific actions and computations to complete its task. Depend on the needed task in specific module we estimated the execution time for each specific module.

- Response Time (RT_t) for specific sending tuple is sum of network delay to execute this tuple (ND_t) and needed execution time of tuple (EXT_t):

$$RT_t = ND_t + EXT_t \qquad (3)$$

- Average execution time (avgEXT) on fog/cloud computed as follow:

$$avgEXT(n) = \frac{EXT_t + n \, . \, avgEXT(n-1)}{n+1} \qquad (4)$$

Where n is the tuple type count and tuple type here means from where this tuple is created. There are three tuple type depend on end user devices which are Reception, Laboratory and Clinic tuple. So, tuple type count is how many tuples are sending from specific type of user. On fog computing each fog server performed only one tuple type which is sending from end user devices that exist at its network boundary such as Reception fog performs and executes only reception tuples. So, in fog computing this equation used to compute average execution time of all tuples in specific fog.

While in cloud computing environment all tuple types executed on cloud. So, this equation is not use to compute average execution time of all tuples in cloud instead that is used to compute average execution time of each tuple type separately. For example, how many tuples received from reception and executed on cloud or how many tuples received from laboratory and executed on cloud and so on.

- Average total execution time on cloud is calculated by dividing the total execution time (TotalEXT) by the number of all executed tuples (TuplesNo) on cloud. This equation used to compute average execution time for all recived tuples in cloud and it doesn’t matter from where this tuples are coming:

$$TotalEXT = \sum EXT_t \qquad (5)$$

$$avgEXT = \frac{TotalEXT}{TuplesNo} \qquad (6)$$

VI RESULTS ANALYSIS

In this subsection, we will show and explain the analysis of collected results. First we collect results for the system validation and then the result of the system performance is collected.

A. System Validation Test

Validation is concerned with checking whether the system will meet the customer’s actual needs or not, in this subsection we will examine the validation of the proposed ABAC. We will check the predefined access policies, if they are applied as customers expect or not. So, we will examine our implemented of ABAC depend on test inputs to check if it meets the required access policies or not.

In our proposed ABAC we assumed the required access policies for each user as following:

1- **Reception** employee can use only reception department computer to view general information of patient EHR for 24 hours.

2- **Phyision** can use only laboratory department computer to view part of patient EHR which are the patient’s general information and required test for 24 hours

3- **Consultant** can use clinic department computer to view part of patient EHR which are the patient’s general information, required test and diagnosis; only deuring work hour (8 am:4 pm).

4- **Consultant** can use clinic department computer to update part of patient EHR which is diagnosis of patient during work hour (8 am:4 pm).

5- **Specialist** can use clinic department computer to view part of patient EHR which are the patient’s general information, required test and diagnosis;

deuring work hour

6- **Nurse** can use clinic department computer to view part of patient EHR which are the patient's general information and diagnosis;for 24 hours

We conducted three different test's cases to validate our ABAC (unauthorized access denied access and authorized access). We considered many test inputs to make sure the access policies for each user is controlled. In our implemented ABAC we have five users (Reception, Physician, Nurse, Consultant, and Specialist) and there are three networks (Reception, Laboratory and Clinic). For each network, there is a fog that received user requests (sending tuples) and applies ABAC on these requests. Each user can only use the devices in its specific network such as the Reception can use only end user devices in reception department to access only Reception fog while the Physician use end user devices in laboratory department to access Laboratory fog. And the Nurse, Consultant and Specialist use only end users devices in clinic department to access Clinic fog.

B. System Performance Test

In this subsection, we will present the analysis of the result of the application first running as fog computing. After that, the analysis of result of the application running as cloud computing. Finally, both results are compared. In each test, we collect the execution time for each sending tuple, response time and average of execution time at fog or cloud server.

1) Selecting the Number of Replications

This section describes chosen method for determining the number of replications that must be performed with a model and for selecting an appropriate run-length for a long run. The aim in both cases is to make sure that enough output data have been obtained from the simulation in order to approximate the model performance with enough accuracy.

A replication is a run of a simulation that uses specific streams of arbitrary numbers. Numerous replications are performed by changing the stream of arbitrary numbers that are re-running the simulation. The question is: how numerous replications need to be performed? There are number of approaches to answer this question as mentioned in [15] and in our test we choose a rule of thumb method to come back with this question.

A rule of thumb Law and McComas (1990) propose that at least three to five replications are performed. This simple rule of thumb is helpful because it makes clear that model users should not rely on the results from a single replication. It does not, but, takes into account the characteristics of a model's output. Models with output data that are very varied normally require more replications than models with a more stable output. Indeed, it is not unusual for a model to need more than five replications before a satisfactory estimate of performance is obtained [15]. Our output data are varied normally so we decided to perform five replications before a satisfactory estimate of performance is obtained.

2) Fog Computing Results

In this subsection we will show collected results and analyse it when the E-health system simulated in fog computing environment. We conduct different experiments and collect its results. And we will analyse every result collected.

a) Execution\Response time of Many Sending Tuples (Unauthorized, Denied or Authorized) on Different Fog

Here we collected the results of different sending tuples which are executed on different fogs. Figure 5 shows the execution time for each unauthorized access which sent to different fogs (reception laboratory and clinic). As shown in the figure the average of execution time for 25 sending tuples on Reception fog is more than the average of execution time on Laboratory and Clinic fog. The Reception fog takes this amount of time because as mentioned in the workflow design chapter, when a patient visits the hospital at the first time his/her EHR will not be available at any hospital department fog. Therefore, Reception fog needs to connect to the cloud to get patient EHR, which increases the time of execution for requests coming from reception as it will take a longer time to get the EHR from cloud. Actually the real execution time on Reception fog (the Reception only forward request from user to cloud) is low but we couldn't say the execution of request is finished until response is received and returned to the user. So, the Reception fog will waits the cloud until it finished request execution and send result back to the reception then it is send to end user. So, execution time of tuple on Reception is sum of execution time of tuple on Reception fog (EXT_r) and execution time of same tuple on cloud (EXT_c) and it is calculated as follow:

$$\mathrm{EXT}_t = \mathrm{EXT}_r + \mathrm{EXT}_c \qquad (7)$$

In our workflow design we assumed and proposed that after this request send to cloud the cloud will send a broadcast of the patient EHR for all hospital's department's fog. Then the copy of patient EHR will be available to use in each hospital’s fog and there is no need to connect to cloud to share EHR. For that execution time of laboratory or clinic tuple if it was unauthorized, denied or authorized will take times less than reception tuple. Also, we noticed that the execution time of unauthorized, denied or authorized access on laboratory and clinic approximately are close to each other. Always reception fog needs time more than laboratory or clinic to execute sending tuples. The three tables below show execution and response time for 25 sending tuples to different fogs (Reception, Laboratory and Clinic).

As shown in fig. 5, 6 and 7 the average of response time is more than the execution time and these results are reasonable as it is mentioned earlier that the response time is equal the execution time of tuples plus the delay of network. In our framework, we estimated this network delay as mentioned previously. The execution time and response time for different sending tuples are different depend on enforced policies. It is noticed from results collected, that unauthorized access always need time less than the time that needed to execute authorized or denied access.

Figure 5 shows collected results of unauthorized access on Reception, Laboratory and Clinic fog.

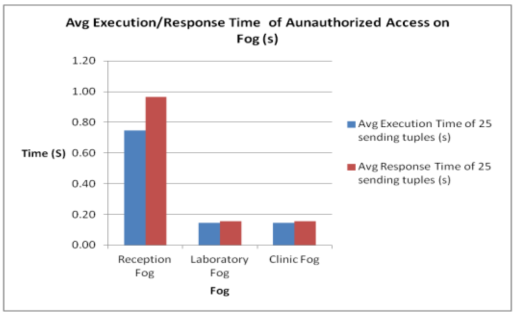

Figure 5. Execution and response time of many sending tuples of unauthorized access

Figure 6 shows collected results of denied access on Reception, Laboratory and Clinic fog.

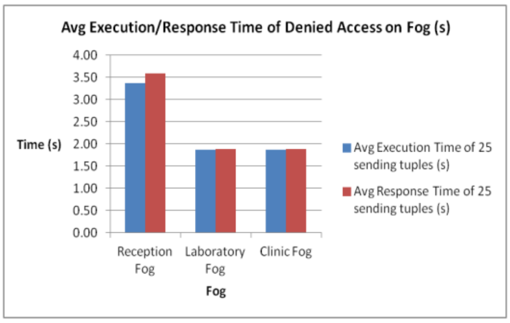

Figure 6. Execution and response time of many sending tuples of denied access

And Figure 7 shows collected results of authorized access on Reception, Laboratory and Clinic fog.

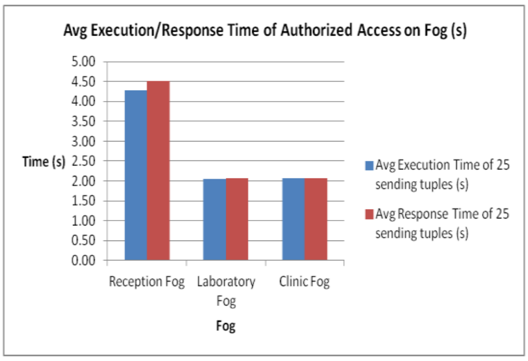

Figure 7. Executions and Response Time of Many Sending Tuples of Authorized Access

b) Average Execution Time of Sending Tuples (Unauthorized, Denie or Authorized) on Different Fog

In addition, the average of execution time is collected 5 times for each case depended on a rule of thumb. Execution time of each tuple for different access (unauthorized, denied or authorized) on different fogs (Reception, Laboratory or Clinic). And from collected results we noticed that unauthorized access take less time than authorized and denied access because the computation on this case is less than computations if the access authorized or denied. The authorized access need more computation as explained in implementation chapter to execute, as it needs to authorize the sending request, get attributes of authorized request and send query, search of policy statement that appropriate for sending query, search of missing attributes if it is available, enforce the policy on authorized request and provide the EHR of patient. As all of these steps have to be executed if the request is authorized, the authorized request takes longer time than other access to execute. On the other hand, denied access request takes more time than unauthorized and less than authorized to execute because it have tasks and computations more than unauthorized but less than authorized. On all fogs the same results appeared for that we choose Laboratory fog arbitrary to present their results as shown in Figure 8.

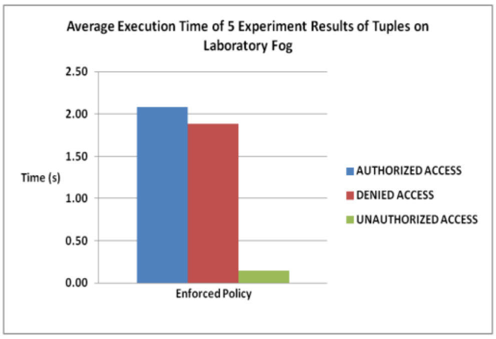

Figure 8. Average execution time results of different sending tuples to laboratory fog

3) *Cloud Computing Results*

In this subsection, we will show collected results and analyze it when the E-health system simulated in cloud computing environment. We collected execution/response time of many sending tuples (Unauthorized, Denied or Authorized) on Cloud. The Figure shows the result of execution time for unauthorized, authorized and denied access which sent to cloud. Unlike fog computing environment there is no a huge difference between execution time of tuple types (reception, laboratory or clinic). Any tuple received on cloud need the same time of execution and it doesn't matter from where this tuple is coming. The differences of execution time of sending tuple in cloud depend on enforced policy. Authorized access take time more than unauthorized and denied to execute because the numbers of modules that are used to perform the request for authorized access is more than modules needed for unauthorized or denied access so the cloud will perform several computations to authorize the access. Also, denied access take a longer time to be executed than unauthorized access because the unauthorized access will return after the authorization check while denied access required more task to be performed and computed.

As shown in Figure . 9, the average of response time is more than the execution time and these results are expected as the response time depends on execution time of tuples plus the delay of network. In our framework, we estimated this network delay as mentioned in implementation chapter.

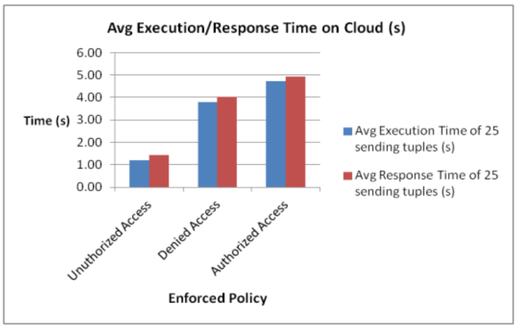

Figure 9. Execution time results of many trial access requests on cloud

4) *Fog and Cloud Computing Comparison*

In this subsection, we will compare the above collected average execution time of sending tuples on fog and cloud and see where the performance of simulated E-health system is better. Figure 0 shows that the execution time which is needed to enforce the access policies on cloud is more than the time that is needed to execute the same enforced policy on fog. In addition, as known the response time is depend on execution time plus network delay and the network delay between end user and cloud is more than network delay between end user and fog -because fog is existing near of network edges (end user devices)-. Moreover, we can derive that the response time of transmitting tuple on fog is also less than the response time on cloud for that the response in fog environment is faster than in cloud computing environment.

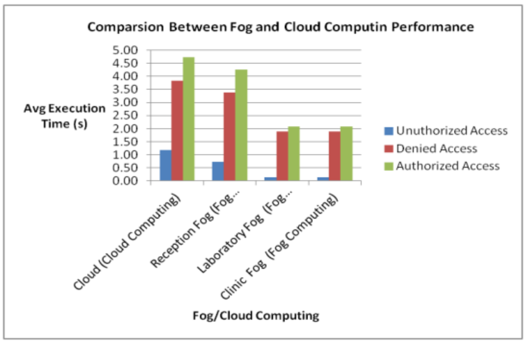

Figure 10. Comparison between average execution time of fog and cloud

VII CONCLUSION AND FUTURE WORK

Cisco recently proposed a new computing environment called fog computing which is a connection of billions of devices (called as fog nodes) around the globe. Cloud computing defers from fog computing in the distribution of processing in distributed nodes with mobility. In fog computing environment, the generic application runs logic on resources throughout the network, including dedicated computing nodes and routers. Fog computing based healthcare solutions can play an important role in improving the quality of health services in near future by supporting low latency during patient's service.

We considered the fog environment as an appropriate platform to deploy and support the Electronic Health Records (EHR). In this research, there were three major contributions first, we simulated E-health system by using iFogSim tool and we came up with iFogSimEhealthSystem simulation tool. Second, we applied ABAC at fog to secure access into EHR of patient. Third, we studied the performance of proposed securing access in E-health system in fog computing.

We considered in our solution the low capabilities of storage and computing of fog nodes by focusing on reducing the storing and processes in fog nodes to serve the availability of fog and to improve its performance and efficiency. The results showed that the performance of fog computing is higher than the cloud computing performance. Moreover, it showed that the response time of request on fog is less than response time if the request sending to cloud. Therefore, if the computing and processing happened near of end user device this will lead to low response

ACKNOWLEDGMENT

I wish to acknowledge my supervisors Dr. Seyed M Buhari and Ahmed Barnawi for their support and advices. Also I would like to thank my family who support me by their prayers and love, and my friend Nora Janbi who encourage and support me.

REFERENCE

[1] Hong K, Lillethun D, Ramachandran U, Ottenwälder B, Koldehofe B, editors. Opportunistic spatio-temporal event processing for mobile situation awareness. Proceedings of the 7th ACM international conference on Distributed event-based systems; 2013: ACM.

[2] Zhu J, Chan DS, Prabhu MS, Natarajan P, Hu H, Bonomi F, editors. Improving web sites performance using edge servers in fog computing architecture. Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on; 2013: IEEE.

[3] Hong K, Lillethun D, Ramachandran U, Ottenwälder B, Koldehofe B, editors. Mobile fog: A programming model for large-scale applications on the internet of things. Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing; 2013: ACM.

[4] Bonomi F, Milito R, Zhu J, Addepalli S, editors. Fog computing and its role in the internet of things. Proceedings of the first edition of the MCC workshop on Mobile cloud computing; 2012: ACM.

[5] Wu R, Ahn G-J, Hu H, editors. Secure sharing of electronic health records in clouds. Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on; 2012: IEEE.

[6] NIST GS, Goguen A, Fringa A. Risk Management Guide for Information Technology Systems. Recommendations of the National Institute of Standards and Technology. 2002.

[7] Boyd C, Mathuria A. Protocols for authentication and key establishment: Springer Science & Business Media; 2013.

[8] Narayanan HAJ, Güneş MH, editors. Ensuring access control in cloud provisioned healthcare systems. 2011 IEEE Consumer Communications and Networking Conference (CCNC); 2011: IEEE.

[9] Jin J, Ahn G-J, Hu H, Covington MJ, Zhang X. Patient-centric authorization framework for electronic healthcare services. computers & security. 2011;30(2):116-27.

[10] Huang J, Sharaf M, Huang C-T, editors. A hierarchical framework for secure and scalable ehr sharing and access control in multi-cloud. 2012 41st International Conference on Parallel Processing Workshops; 2012: IEEE.

[11] Cao Y, Chen S, Hou P, Brown D, editors. FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation. Networking, Architecture and Storage (NAS), 2015 IEEE International Conference on; 2015: IEEE.

[12] Mei B, Cheng W, Cheng X, editors. Fog Computing Based Ultraviolet Radiation Measurement via Smartphones. Hot Topics in Web Systems and Technologies (HotWeb), 2015 Third IEEE Workshop on; 2015: IEEE.

[13] Gia TN, Jiang M, Rahmani A-M, Westerlund T, Liljeberg P, Tenhunen H, editors. Fog Computing in Healthcare Internet of Things: A Case Study on ECG Feature Extraction. Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on; 2015: IEEE.

[14] A.V. Dastjerdi, H.Gupta, S.K. Ghoshy, and R.Buyya,"iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments, 2016; 00:1–22. arXiv:1606.02007v1 [cs.DC] 7 Jun 2016

[15] Stewart Robinson, Simulation: The Practice of Model Development and Use. Chichester, England: John Wiley & Sons Ltd, 2004.