IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 20, 2018, accepted June 23, 2018, date of publication July 18, 2018, date of current version September 21, 2018.

*Digital Object Identifier 10.1109/ACCESS.2018.2856896*

# Privacy-Aware Efficient Fine-Grained Data Access Control in Internet of Medical Things Based Fog Computing

**XIAOFAN WANG[1], LEI WANG[1], YUJUN LI[2], AND KEKE GAI[3]**

[1]School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China
[2]School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China
[3]School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Xiaofan Wang (wangxfok@xaut.edu.cn)

**ABSTRACT** The recent development of cloud computing has empowered the Internet-based services, which enable users to gain a broad scope of access to their applications, such as Internet of Medical Things (IoMT). Considering the efficiency performance, the privacy protection is often kept at a lower level in order to ensure the application can offer a higher level performance. However, this mechanism also causes a serious concern of privacy hazards due to the information over collections operated by apps/applications. Addressing this privacy issue, this paper proposes an approach that is designed to provide high-level privacy protection without lowering down the efficiency in cloud/fog computing (especially in biological systems of IoMT). The proposed model is called fog-based access control model. The fine-grained data access control is combined with the implementation of fog computing in the proposed approach. Our simulation experiment has shown that our approach could offer high-level privacy protection within a shortened execution time, thus it can be useful for IoMT-based applications.

**INDEX TERMS** Internet of Medical Things, fog computing, privacy-aware, fine-grained data access control, privacy time optimization.

## I. INTRODUCTION

Recent rapid development of the Web-based technology has widely driven a remarkable growth of various network-enabled solutions, such as *Internet-of-Things* (IoT), cloud computing, and edge/fog computing [22]. As a presentation of the advanced-level cloud computing, fog computing further powers the functions of cloud solutions by adding an additional server layer between edge devices and remote servers in clouds. The enhanced layer has a perfect matching with other Web-oriented techniques, such as IoT or *Cyber-Physical Systems* (CPSs). IoT has been proposed as a promising means to greatly improve the efficiency and quality of patient care. Medical devices in healthcare IoT termed as Internet of Medical Things (IoMT) measure patient's vital signs and aggregate these data into medical files which are uploaded to the cloud for storage and access by healthcare workers. The IoMT is the collection of medical devices and applications that connect to healthcare IT systems through online computer networks.This emerging computing service model has brought at least two major advantages, even though some other benefits are also relevant to using fog computing.

The first advantage of implementing fog computing is to reduce communication costs that are caused by the high volume of the data transmission.Currently, numerous edge devices are collecting a large number of data from the edge device side and the speed of the data volume growth is even turning into faster along with the increasing amount of the edge nodes. Direct connections between edge devices and cloud servers result in a heavy communication workload, which generally restrict the performance and implementation of many current distributed computing solutions. The other advantage is that implementing fog computing makes it possible to deploy dense geographical distributions with high performance [7], [10], [20]. Workloads can be distributed and processed according to real-time conditions or requirements due to multiple processing options, at least including edge devices, fog servers, and cloud servers. Thanks to the flexible deployment, system designers do not have to rely on a dense

and complicated networking setting in a certain geographical area.

Despite observable advantages deriving from fog computing, the hazard of adversarial activities is not avoided and the security and privacy concerns still restrict its implementations [6], [14], [21]. Those threats that are caused from using the Internet always attach to the hazards. Some threats are becoming more serious in the mobile computing context due to various reasons, such as the variety of user accesses and unencrypted data transmissions. For instance, some mobile apps over collect data from mobile ends without notifying users or data owners. Some literatures [19] have discussed the phenomenon and stated that most mobile apps had more than 7 permission requests, some of which carried privacy. The collected data are transferred in an unencrypted manner. Privacy can be released once adversarial actions take place on mobile devices. Thus, finding out an effective approach of securing big volume of data is urgent. As one of the common data protection methods, a competent data access mechanism is expected to be an potential solution to the security issue above.

In this paper, we develop a new approach that deploys an additional control layer at fog servers in order to increase security protection capability for regional mobile devices. The proposed model is called *Fog-based Access Control Model* (FACM). Fig. 1 shows the high level architecture of the proposed model that illustrates a basic deployment. As shown in the figure,

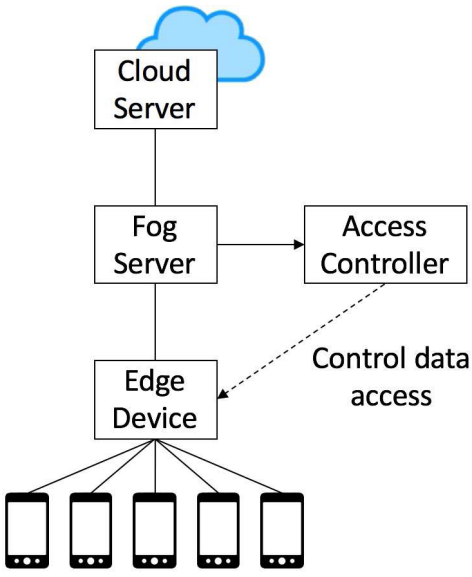


**FIGURE 1.** High level architecture of the proposed model.

The main contributions of this work are twofold that are presented as follows.

1) This work proposes a fine-grained access control mechanism considering the implementation of the cloud-based solution in either mobile or non-mobile context. The main novelty of our approach is using an additional layer of the fog server, in which customized access control setting can be provided.
2) Our approach is suitable for various implementation scenarios, which include data storage, directories, and file management. The establishment of our approach is feasible for most users due to its simple structure and interface. Considering the efficiency of the popularity, our approach only needs limited changes on the current active systems.

The rest of this paper is structured as the following order. First, Section II provides a literature review addressing the recent work of fog computing for the theoretical support. Next, Section III formulates the proposed security problem into a solvable optimization problem, so that the proposed model is described in this section. Moreover, we explain our proposed algorithm in Section IV and show evaluations in Section V. Finally, a conclusion is drawn in Section VI.

## II. RELATED WORK

Some studies have attempted to design fine-grained data protection methods in the mobile context over years. For example, an approach using an attribute-based mechanism was proposed to secure data from setting up data owners' rules [9]. The design of the attribute-based data protection method generally depended on the role who was required to be protected. Essential approaches used for protecting a role included defining secure policies and making secure rules [2]. An advanced attribute-based secure mechanism might involve multi-layer or multi-access schemes [8], [11], [18], [27]. However, the main drawback of using an attribute-based approach was that the implementation could bring the high workload to the user side. In general, the complexity of the method for protecting the target role, such as policies and rules, was tied to the level of the security. For most common mobile users, creating a complicated data access mechanism was a tough job due to the limited knowledge in rule/policy designs.

Next, data owners generally did not have any control on their data once the data were transferred to the cloud side. This fact also weakened the performance of the attribute-based security method [30]. For those data carrying sensitive information, they might be abused by unexpected parties without noticing data owners. It implied that designing a fine-grained access control had a chance to reduce the chance of the privacy leakage [12]. Prior work had explored a variety of approaches in designing fine-grained access control methods. For example, combing attribute-based schemes with the fine-grained mechanism was an option [29]. The motivation of developing this type of approach was that the level of the trust was varied for cloud servers and data owners.

Meanwhile, fog computing has been introduced to the computing world as an intensifier for cloud-based applications [3], [23]. Distributed computing system is becoming remarkably complicated as the number of the connected devices as well as servers has been dramatically grown. Despite of many similarities between cloud and fog

computing in storage and application, fog computing distinguishes from cloud computing in the aspect of geographical distributions. Implementing fog computing drags a centralization-based solution (cloud) to a mixture mode that consists of both centralized and distributed computing [5], [15]. It enables a distributive workload assignment to reach the nearest computing resource, which intends to save costs, such as lightening volume of the communication, lowering down computations, or less data transmissions. Fog servers play an important role in this advanced system, since it arranges, controls, and assigns incoming workloads to the proper computation nodes or storage space. The setting of fog computing makes it possible to have a capability of the layered control [13].

One of the crucial security concerns in fog computing is that some fog servers become a vulnerability for the system [25]. Since a fog server has a certain control ability over edge devices and connects to the remote cloud servers, attacking this intermediate layer will be an efficient way for attackers. Some prior studies tried to use radio access networks such as using multiple spectrum channels to achieve device-to-device adjustable access controls [28]. Some scholars explored the security solutions from other perspectives, such as utilizing the advantage of the resource management [1] and designing privacy-aware scheduling techniques [4]. Recently, fog computing is used in medical applications for bio-signals optimization [24], [26]. Moreover, data privacy and security in healthcare scenarios [16], [17], do not focus on the system security improvement; rare of which had addressed the access control improvement at the fog server layer.

In this work, the vital contribution is novel since the exploration standpoint is different from most prior studies, based on our literature review. We utilize fog servers to add an additional access control layer, so that it avoids changing the basic structure of fog computing and fully utilize the benefit of the fog in the aspect of task scheduling.The flowchart of system model is shown in Fig2. The following section will detailed introduce the mechanism of our approach.

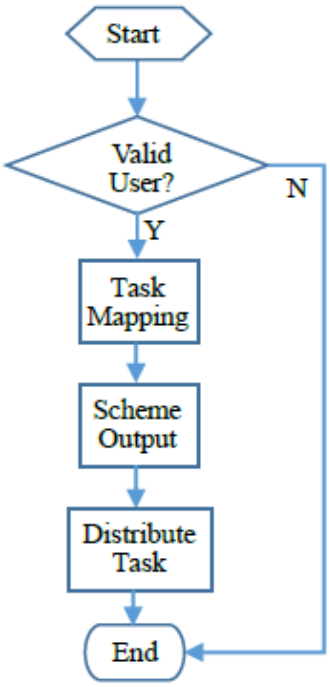


**FIGURE 2.** Flowchart of system model.

## III. PROBLEMS FORMULATION AND MODEL DESIGN

In this section, we mainly clarify the problem that we aim to solve and describe our solution. More specifically speaking, Section III-A further explains the existing struggling issues and formulates the problem of the access control issue into an optimization problem. Section III-B shows the design and mechanism of our solution.

### A. PROBLEM FORMULATION

The main goal of our proposed approach is to reduce the threat caused by over collections that are operated by cloud-based apps. It determines that the crucial issue is the permission authorization that is currently configured by the operating system in most default situations. The contemporary common setting generally is coarse-grained and is hard to be changed for most users. Mobile apps will obtain access authorizations to a large scope of data stored on the mobile user side once the permission agreement is processed. Unfortunately, the operating system is not responsible for setting up a customized access control so that it leaves users with no choice. It allows an app/application to have a full access to the data according to one-switch style agreement.

For example, an end user needs to upload his/her personal data to a social networking site. In the current operating context, a full permission of the authorization needs to be issued to the social networking site. The user either approves a full permission or denies the permission and gives up the data upload operation. Theoretically speaking, once the permission is issued, the site will permanently keep all authorizations until the user takes back the permission or uninstall the app. In fact, considering the usage of the data, the social network is collecting more data than it really needs for providing services. A full permission implies that it has all authorizations to access to sensitive data, as same as other apps/applications with the same permission. It raises the risk of the privacy leakage as each app/application can be an adversarial target and privacy can be stolen once one target is taken.

Thus, seeking out a flexible and feasible approach that can successfully configure access control configuration is our target. We also define the meanings of the flexibility and feasibility in our approach. First, term *Flexibility* refers to that the access control technique is adjustable at the fog server layer. The adjustment can be processed by either system administrators or end users, which depends on the demand of the application. Next, *Feasibility* refers to that the cost of

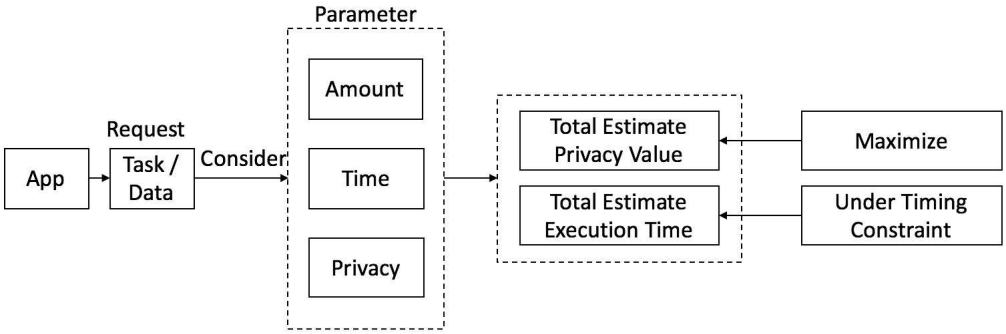


**FIGURE 3.** Diagram of the problem formulation.

implementing the proposed approach is effective low when ensuring the enhancement of the privacy protection. Setting up a timing constraint as a tolerant scope is the method of ensuring efficiency in our approach.

Moreover, in order to make the issue solvable, we formulate the problem into an optimization format. The proposed problem in this work considers a few elements, which formulates a multi-dimensional access control governance. The first element is the volume of the input tasks, which mainly refers to the number of the task and the size of the each task. It generally determines the size of the input. The next element is the property of the task that includes all parameters determining the task feature. Discerning the input task is a fundamental for creating a security strategy for access control. Some typical parameters include the metadata for each task, privacy label, and task type. Meanwhile, our proposed problem also considers the efficiency of the system, so that time is another vital parameter in the process of the strategy-making.

In the following Definition 1, we provide a definition of our proposed problem that is called a **Privacy-Time Optimization** (PTO) problem. We define the inputs, outputs, and objective of the problem, accordingly.

*Definition 1 (Privacy-Time Optimization (PTO) Problem:) Inputs include a set of input tasks* $\{T_i\}$*, a set of work modes* $\{M_j\}$*, the number of the input task* $N^T$*, the timing length of processing each task at jth work mode* $\mathcal{T}_{i,j}$*, the privacy estimate value for each task at jth work mode* $P_{i,j}$*.*

*The output will be a scheduling strategy that determines the access control method for each task,* $\mathbb{S}$*.*

*The objective is to find out a strategy that can obtain an optimal solution to maximize the total value of the estimate privacy under a timing constraint TC.*

As described in Definition 1, we configure that main inputs include the information of the input data that are considered the input tasks. The information includes a number of parameters that can influence the strategy-making of the access control, such as the size of the data, privacy level estimates, time length for each working mode, and other relevant computing costs. The reason for considering these parameters is for making the available access control modes comparable. The main output is a strategy for determining fine-grained access control that considers both total estimate privacy value and total estimate execution time. Fig. 3 shows a diagram of the proposed problem formulation.

According to the statement given in Definition 1, we also provide a mathematical expression about the problem formulation. Assume that there exists a function $f(x)$ and its value scope is within [1, n]. We use the function $f(x)$ to represent the selection of the access control mode. For example, $f(i) = j$ represents that the task $T_i$ selects the j*th* access control mode. We show the calculation of the total privacy estimate value, $\mathbb{P}$, in Eq. (1):

$$\mathbb{P} = \sum_{i=1}^{N^T} P_{i,f(i)} = P_{1,f(1)} + P_{2,f(2)} + \cdots + P_{n,f(n)} \quad (1)$$

Using the same function $f(x)$ can obtain the calculation method for summing up the execution time. Eq. (2) shows the equation of calculating the sum of the time.

$$\mathcal{T}_{total} = \sum_{i=1}^{N^T} \mathcal{T}_{i,f(i)} = \mathcal{T}_{1,f(1)} + \mathcal{T}_{2,f(2)} + \cdots + \mathcal{T}_{n,f(n)} \quad (2)$$

Our goal is to maximize the value of $\mathbb{P}$ while ensuring $\mathcal{T}_{total} \leqslant TC$. It can be formulated by Eq. (3):

$$\begin{cases} \text{MAX}[\mathbb{P}] \leftarrow f(x) \\ \mathcal{T}_{total} \leqslant TC \end{cases} \quad (3)$$

The next section shows the system design of our approach.

### B. SYSTEM DESIGN

In our approach, we consider two types of the data in the privacy creation, which are permanent and temporary data. The permanent data refers to the data carrying the lasting information to the object, such as file property, image, and file creation time. The temporary data refers to some data carrying dynamically changing information, such as geographical locations, access frequency, and other updatable data.

*Access Controller:* We design an entity located at the fog server, which is called *Access Controller* (AC) shown in Fig. 1. Implementing AC is a crucial component of the system. The proposed AC is an operation process for deciding

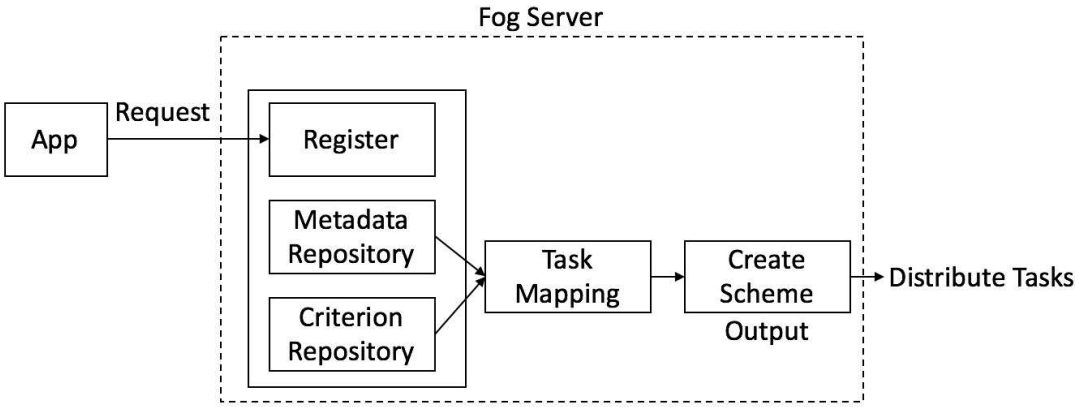


**FIGURE 4.** Secure mechanism design for Access Controller in fog servers.

the fine-grained access control strategy, which uses the data repositories stored in fog servers. Fig. 4 shows the mechanism of the AC, which is marked in the broken lines box in the figure. Three key parts in an AC include *Register*. *Metadata Repository* (MR), and *Criterion Repository* (CR).

A register in AC is responsible for communicating with apps/applications. Data access requests are sent to the AC in which the registration status can be checked. The register will confirm whether the incoming task has been registered in the past. A comparison will be processed when the task type is registered in order to make sure the privacy setting is updated. In essence, the communication needs three typical information, including service request content, app, and data.

The service request information involves a request identity, a time stamp stored the time/date of the request, and the privacy value settled by users. The privacy value that is made by users has the highest priority while making access control strategy. We configure that there are a few levels of the privacy estimate, denoted by a group of integers [1, n]. ''1'' means the lowest value and ''n'' means the highest settled value. Each level is associated with a privacy reward value that is used for calculating the total sum of the estimate privacy value.

Next, the app information involves app header (name, identity) and a trust level value settled by users. A trust level value refers to the level of the trustworthy decided by the user. This is a reference parameter during the access control strategy-making. It will have a lower impact than that of privacy value since users may have a limited understanding on the app/application. Meanwhile, the data information involves data types, type and size. The privacy level is associated with the service request.

Moreover, the governance of the metadata is for information collections. We read metadata and separately store them in an individual folder. An MR is not a shareable space, in which metadata are stored for the purpose of the analysis. The method of tying up metadata and the content is to use identity number to connect both metadata and the content. All the operations above are required for creating a fundamental of forming inputs.

The implementation of MR also is associated with CR. As shown in Fig. 4, working together MR and CR leads to a task mapping. The function of the CR is to store various privacy level settings. The stored criterion maybe dynamically changing, since the setting configured by users can be continuously varied during different time periods. The CR updates the criterion after each comparison for future uses. The criterion may consist of a number of levels, which determines the granularity of the data access controls. Table 1 shows an example of the criterion table that consists both pre-stored privacy estimate values and the request privacy estimate values. The table illustrates three levels for the privacy estimates in this case. More levels of the privacy setting will bring a higher-level fine-grained access control. Eventually, a task mapping can be created from reading service requests and comparing with the existing criterion. The task mapping covers both data storage at AC and data index in the remote cloud server. Information at AC shall be synchronized with the information in the clouds.

**TABLE 1.** Example of the criterion table. (PPE: pre-stored privacy estimate; RPE: request privacy estimate).

| App | PPE | RPE | Task Type |
|---|---|---|---|
| Message | 1 | 2 | Contact |
| Message | 1 | 1 | Image |
| Message | 3 | 3 | Text |
| Contact | 3 | 3 | Name |
| Contact | 1 | 2 | Telephone |
| Email | 2 | 3 | Email Address |
| Email | 1 | 2 | Image |
| Email | 1 | 3 | Contact Name |

The last step of the AC is to create a scheme for access controls. In our approach, a two-step greedy algorithm has been proposed. Implementing the proposed algorithm will produce an updated access control strategy that covers both privacy protection and execution efficiency. The next section will introduces our main algorithms used in the approach.

## IV. PROPOSED ALGORITHMS

We propose an algorithm named *Access-Control Determination* (ACD) algorithm to support the decision-making on access controls. The purpose of this algorithm is to create a fine-grained access control mechanism, which can maximize the total privacy estimate values while meeting the requirement of the timing constraint. The implementation takes place at the access controller.

As mentioned in Section III-A, inputs of this algorithm include a set of input tasks $\{T_i\}$, a set of work modes $\{M_j\}$, the number of the input task $N^T$, the timing length of processing each task at $j$th work mode $\mathcal{T}_{i,j}$, the privacy estimate value for each task at $j$th work mode $P_{i,j}$. A pre-stored criterion table $\mathbb{C}$ is needed. The criterion table can be represented by a 2-tuple $\langle ID_c, M_c \rangle$ so that $ID_c$ refers to the file identity and $M_c$ refers to the stored privacy level. The output will be a method guiding access controls, $\mathbb{S}$. Similar to the criterion table, each access control can be denoted by a 2-tuple $\langle ID, M \rangle$, in which $ID$ means the task identity and $M$ refers to the access control mode.

Pseudo codes of the ACD algorithm is shown in Algorithm 1. Major steps of this algorithm include:

1) Read all inputs including task parameters and the criterion table. A set of 2-tuple for access control determinations needs to be initialized.
2) The first step of this algorithm needs to check the difference between incoming tasks and pre-stored information. The pre-stored privacy level table needs to be updated according to the user's configuration. Once the file is found as a registered objective but the content is varied, the information in the table will be updated; otherwise, the information in the table will be kept without any change. When the file is not found in the table, a new file will be created according to the user's setting.
3) The next step is to assign the privacy-aware access control working modes. As mentioned in the previous section, the highest privacy protection setting made by users must be processed first. Thus, a limited authentication permission is given to those task with the highest expectation of the privacy protection before considering other tasks. We sort all tasks according to their privacy levels in a descending order to complete this step. Add all assigned tasks' privacy estimate values to the strategy table. Update the timing constraint by subtracting
4) For the remaining tasks, we need to consider the execution time length, since we have a timing constraint. We sort the remaining tasks according to the execution time in an ascending order. At this step, we aim to increase the number of the fine-grained access controls in order to increase the total privacy estimate value. The operation will be ended when there is no time left. Update the strategy table accordingly.
5) Return the final strategy table.

**Algorithm 1 Access-Control Determination (ACD) Algorithm**

**Require:** $\{T_i\}$, $\{M_j\}$, $\mathcal{T}_{i,j}$, $\mathbb{C}$
**Ensure:** $\mathbb{S}$

```
Read all inputs {T_i}, {M_j}, T_{i,j}, C, initialize S
for ∀ tasks (data) do
    if File is already registered and difference is found then
        Update the file in C
    else if File is already registered and no difference is found then
        Keep the file in C
    else if File is unregistered then
        Create the file in C
    end if
end for
for ∀ tasks do
    Sort tasks according to the privacy levels
    if Privacy configuration is the highest (configured by users) then
        Assign the lowest authorization
        Calculate the time cost and update the remaining time
        Update S
    else
        for ∀ tasks left do
            Sort the tasks according to the execution time
            if task time is shorter than the remaining time then
                Assign the access control level accordingly
                Update the time
            else
                Assign full permission as the original system setting
            end if
        end for
        Update S
    end if
end for
return Final S
```

The design of ACD algorithm mainly considers two aspects. The first aspect is to make sure that the information with the high privacy protection expectation needs to be added to a fine-grained access control list first. It ensures the intention of users that protect their privacy by a lower-level authentication for apps/applications. The other aspect is that we target at having a higher-level granularity while satisfying the timing constraint. Our method is that we give the access control mode with a shorter execution time a priority. The next section presents the evaluation results to prove the performance of our approach.

## V. EXPERIMENT EVALUATIONS

In this section, we present the assessment of our FACM in order to examine its performance. Section V-A provides a brief introduction about the experiment settings used in the assessment. Section V-B shows a few experiment results collected from our evaluations.

### A. EXPERIMENT SETTINGS

Our experiment setting was designed for examining the performance of the proposed approach. We developed a simulation tool (a Java program) to investigate our approach and estimate its performance. The aspects of the examination were twofold. The first aspect was to examine the performance in maximizing the total privacy estimate value. In order to observe the difference between our approach and optimal solution, we used a *Brute Force* (BF) algorithm to obtain optimal solutions. The other aspect in our evaluations was examining the execution time for creating the strategy. The purpose of this examination was to ensure our approach could provide a real-time service.

The hardware configuration included an AMD A8-7410 accelerated processor, an 8GB system memory, and 1TB hard drive. Operating system is a Windows 10 (Microsoft). The examinations were implemented on our simulator that was developed by Java.

We implemented a variety of experiment settings to simulate different applications. Partial experiment settings included:

1) Setting 1: The number of the input task: from 1 to 8; the execution time distribution range: from 1 to 10; the range of the privacy level estimates: from 1 to 5.
2) Setting 2: The number of the input task: from 1 to 8; the execution time distribution range: from 1 to 20; the range of the privacy level estimates: from 1 to 6.
3) Setting 3: The number of the input task: from 5 to 10; the execution time distribution range: from 1 to 30; the range of the privacy level estimates: from 1 to 3.

The next section displayed partial experiment results.

### B. EXPERIMENT RESULTS

Fig. 5 - Fig. 7 presented partial evaluation results about the average execution time of implementing our approach. Each round evaluations were based on a 10-time collection so that the values were average statistics. Fig. 5 showed the results collected from experiment setting 1, which were based on a 5-round evaluation. The scope of the execution time was from $4 \times 10^4$ to $10 \times 10^4$ nanoseconds.

Fig. 6 showed the results collected from experiment setting 2, which were based on a 10-round evaluation. The scope of the execution time was from $3 \times 10^4$ to $8 \times 10^4$ nanoseconds, which was similar to the results obtained from setting 1. Meanwhile, Fig. 7 showed the results gathered from experiment setting 3. The number of the experiment round was as the same as that of setting 2. We saw that the scope of the execution time was from $3 \times 10^4$ to $15 \times 10^4$, which was broader than two other settings. This consequence implied that the execution time length was associated with the number of the input task. We also observed that the time lengths under these three settings were all acceptable. The implementation was efficient enough for delivering a real-time service.

In addition, Fig. 8 - Fig. 10 showed partial results collections for total privacy estimates, which compared our

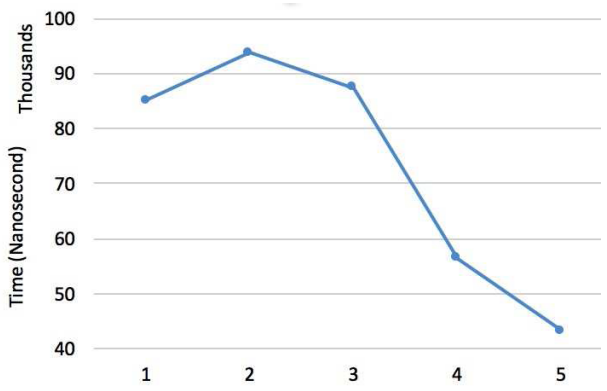


**FIGURE 5.** Partial result collections for average execution time from 5 rounds evaluations under Setting 1.

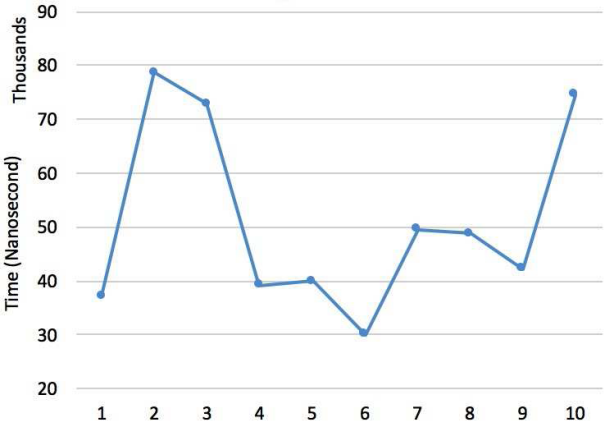


**FIGURE 6.** Partial result collections for average execution time from 10 rounds evaluations under Setting 2.

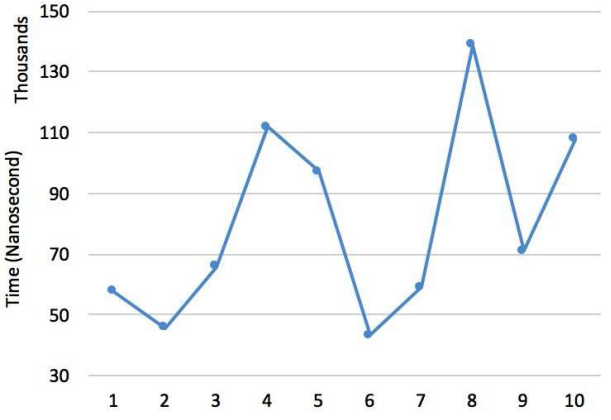


**FIGURE 7.** Partial result collections for average execution time from 10 rounds evaluations under Setting 3.

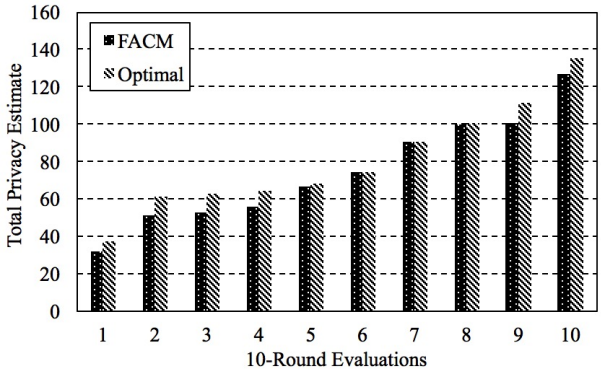


**FIGURE 8.** Partial result collections for total privacy estimates comparing FACM with the optimal solution under Setting 1.

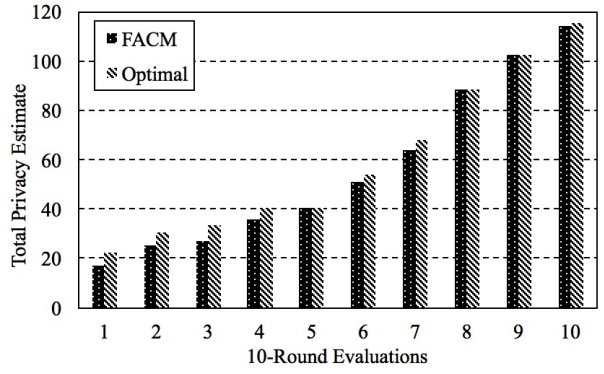


**FIGURE 9.** Partial result collections for total privacy estimates comparing FACM with the optimal solution under Setting 2.

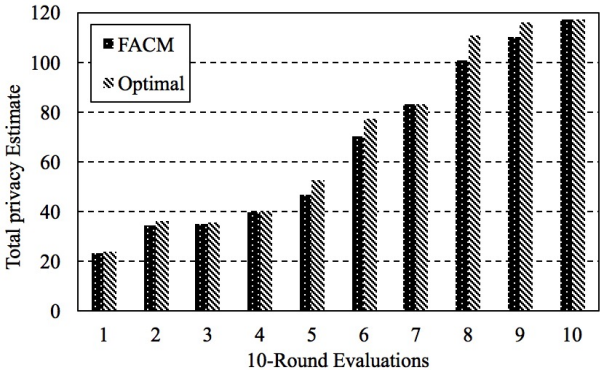


**FIGURE 10.** Partial result collections for total privacy estimates comparing FACM with the optimal solution under Setting 3.

approach with optimal solutions obtained from the BF algorithm. The values of the privacy estimates were counted in units. The evaluation results were sorted in an ascending order in order to make the difference observable. We used the following formulation to examine the difference (*Diff*) between our approach and the optimal solution:

$$Diff = (Op - FACM)/FACM$$

In the equation, *Diff* refers to the difference rate; *Op* refers to the optimal solution of the total privacy estimates; *FACM* refers to the total privacy estimate values obtained from our approach.

According to our statistics, the average of *Diff* under setting 1 was 9.51%; the average of *Diff* under setting 1 was 9.94%; the average of *Diff* under setting 1 was 5.04%. These results depicted that our approach could perform a near-optimal solution in the given application scenarios. Considering individual assessment, we could see that there was slight difference for each pair bars in figures. The results showed that our model could nearly maximize the privacy protection value by utilizing the available computing resource. Since the value of the estimate privacy implied the level of the data protection, it also proved that the amount of the data carrying privacy could be increased and strengthened by implementing the access controller in our model. Moreover, Fig. 9 and Fig. 10 also showed similar results to Fig. 8, which further evaluate the correctness of implementing our model from the perspective of the privacy protection.

In summary, our evaluations had shown that our approach was adoptable in practice from the perspectives of security and efficiency. The total privacy estimate value could reach a near-optimal solution level; the execution time of creating strategies was effectively short.

### C. DISCUSSIONS

Along with patients' privacy, medication data security also is one of the significant requirements for establishing a secure IoMT. It is another side that will be addressed in our future work. In fact, our model can efficiently not only protect users' privacy but also increase the data security, from the theoretical perspective. The level of the data security can be enhanced by configuring the method of the data encryption. In our future work, we will further expand our model by providing more encryption options and enable those data carrying various levels of the privacy to be protected by different encryption methods. In general, the level of security protection will be associated with the level of the privacy.

## VI. CONCLUSIONS

This paper presented a novel mechanism that could successfully reduce risks of the privacy leakage for IoMT based applications. The proposed mechanism used an additional layer between edge users and cloud servers to set up the customized access control. The implementation located at the fog layer in which function-oriented servers could provide each app with a demanded access control service.

## REFERENCES

[1] M. Aazam and E. Huh, "Dynamic resource provisioning through fog micro datacenter," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, St. Louis, MO, USA, Mar. 2015, pp. 105–110.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, Berkeley, CA, USA, Mar. 2007, pp. 321–334.

[3] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.

[4] M. Du, K. Wang, X. Liu, S. Guo, and Y. Zhang, "A differential privacy-based query model for sustainable fog data centers," *IEEE Trans. Sustain. Comput.*, to be published, doi: 10.1109/TSUSC.2017.2715038.

[5] M. A. A. Faruque and K. Vatanparvar, "Energy management-as-a-service over fog computing platform," *IEEE Internet Things J.*, vol. 3, no. 2, pp. 161–169, Apr. 2016.

[6] K. Gai and M. Qiu, "Blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers," *IEEE Trans. Ind. Inform.*, vol. 14, no. 8, pp. 3590–3598, 2018.

[7] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," *J. Netw. Comput. Appl.*, vol. 59, pp. 46–54, Jan. 2016.

[8] S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters, "Attribute-based encryption for circuits from multilinear maps," in *Advances in Cryptology—CRYPTO*. Santa Barbara, CA, USA: Springer, 2013, pp. 479–499.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, Alexandria, VA, USA, 2006, pp. 89–98.

[10] N. Iotti, M. Picone, S. Cirani, and G. Ferrari, "Improving quality of experience in future wireless access networks through fog computing," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 26–33, Mar./Apr. 2017.

[11] A. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," in *Advances in Cryptology—CRYPTO*, vol. 7417, Santa Barbara, CA, USA: Springer, 2012, pp. 180–198.

[12] J. Li, X. Chen, J. Li, C. Jia, J. Ma, and W. Lou, "Fine-grained access control system based on outsourced attribute-based encryption," in *Proc. Eur. Symp. Res. Comput. Secur.*, Egham, U.K.: Springer, 2013, pp. 592–609.

[13] T. Li, Y. Liu, L. Gao, and A. Liu, "A cooperative-based model for smart-sensing tasks in fog computing," *IEEE Access*, vol. 5, pp. 21296–21311, 2017.

[14] H. Madsen, B. Burtschy, G. Albeanu, and F. Popentiu-Vladicescu, "Reliability in the utility computing era: Towards reliable fog computing," in *Proc. 20th Int. Conf. Syst., Signals Image Process.*, Bucharest, Romania, 2013, pp. 43–46.

[15] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 46–53, Jul./Aug. 2016.

[16] S. Pirbhulal, H. Zhang, S. Mukhopadhyay, W. Wu, and Y. T. Zhang, "An efficient biometric-based algorithm using heart rate variability for securing body sensor networks," *Sensors*, vol. 15, no. 7, pp. 15067–15089, 2015.

[17] S. Pirbhulal, H. Zhang, S. Mukhopadhyay, W. Wu, and Y. T. Zhang, "Heart-beats based biometric random binary sequences generation to secure wireless body sensor networks," *IEEE Trans. Biomed. Eng.*, to be published, doi: 10.1109/TBME.2018.2815155.

[18] M. Qiu, K. Gai, B. Thuraisingham, L. Tao, and H. Zhao, "Proactive user-centric secure data scheme using attribute-based semantic access controls for mobile clouds in financial industry," *Future Generat. Comput. Syst.*, vol. 80, pp. 421–429, Mar. 2016.

[19] C. Rottermanner, P. Kieseberg, M. Huber, M. Schmiedecker, and S. Schrittwieser, "Privacy and data protection in smartphone messengers," in *Proc. 17th Int. Conf. Inf. Integr. Web-Based Appl. Services*, Brussels, Belgium, 2015, p. 83.

[20] Y.-Y. Shih, W.-H. Chung, A.-C. Pang, T.-C. Chiu, and H.-Y. Wei, "Enabling low-latency applications in fog-radio access networks," *IEEE Netw.*, vol. 31, no. 1, pp. 52–58, Jan. 2017.

[21] S. Stolfo, M. Salem, and A. Keromytis, "Fog computing: Mitigating insider data theft attacks in the cloud," in *Proc. IEEE Symp. Secur. Privacy Workshops*, San Francisco, CA, USA, 2012, May pp. 125–128.

[22] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014.

[23] J. Wang, J. Cao, B. Li, S. Lee, and R. S. Sherratt, "Bio-inspired ant colony optimization based clustering algorithm with mobile sinks for applications in consumer home automation networks," *IEEE Trans. Consum. Electron.*, vol. 61, no. 4, pp. 438–444, Nov. 2015.

[24] J. Wang, Z. Zhang, B. Li, S. Lee, and R. S. Sherratt, "An enhanced fall detection system for elderly person monitoring using consumer home networks," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 23–29, Feb. 2014.

[25] Y. Wang, T. Uehara, and R. Sasaki, "Fog computing: Issues and challenges in security and forensics," in *Proc. 39th Annu. Comput. Softw. Appl. Conf.*, vol. 3, 2015, pp. 53–59.

[26] W. Wu, S. Pirbhulal, A. K. Sangaiah, S. C. Mukhopadhyay, and G. Li, "Optimization of signal quality over comfortability of textile electrodes for ECG monitoring in fog computing based medical applications," *Future Gener. Comput. Syst.*, vol. 86, pp. 515–526, 2018.

[27] F. Xiao, W. Liu, Z. Li, L. Chen, and R. Wang, "Noise-tolerant wireless sensor networks localization via multinorms regularized matrix completion," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2409–2419, Mar. 2018.

[28] S. Yan, M. Peng, and W. Wang, "User access mode selection in fog computing based radio access networks," in *Proc. IEEE Int. Conf. Commun.*, Malaysia, May 2016, pp. 1–6.

[29] K. Yang, X. Jia, and K. Ren, "Attribute-based fine-grained access control with efficient revocation in cloud storage systems," in *Proc. 8th ACM SIGSAC Symp. Inf., Comput. Commun. Secur.*, Hangzhou, China, 2013, pp. 523–528.

[30] Y. Zhu, H. Hu, G. Ahn, M. Yu, and H. Zhao, "Comparison-based encryption for fine-grained access control in clouds," in *Proc. 2nd ACM Conf. Data Appl. Secur. Privacy*, San Antonio, TX, USA, 2012, pp. 105–116.

**XIAOFAN WANG** received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2003, respectively, and the Ph.D. degree in computer science and engineering from Xidian University, Xi'an, in 2012. He is currently an Associate Professor with the Faculty of Computer Science and Engineering, Xi'an University of Technology. His current research interests focused in data mining, machine learning, and intelligent information processing.

**LEI WANG** received the B.S. and M.S. degrees in computer science and technology from the Xi'an University of Technology, Xi'an, China, in 1994 and 1997, respectively, and the Ph.D. degree in electronic science and technology from Xidian University, Xi'an, in 2001. He is currently a Professor with the Faculty of Computer Science and Engineering, Xi'an University of Technology. His current research interests include evolutionary algorithms, neural networks, and data mining.

**YUJUN LI** received the B.E. and M.E. degrees from the Xi'an University of Technology, Xi'an, China, and the Ph.D. degree from Xi'an Jiaotong University, Xi'an. He is currently a Lecture with the Xi'an University of Technology. His research interests include spectrum analysis, smart sensor, statistical modeling, and optimization.

**KEKE GAI** received the Ph.D. degree from the Department of Computer Science, Pace University, NY, USA. He is currently an Associate Professor with the Institute of Technology, Beijing. His research interests include cloud computing, network security, network communication.