An Optimization Framework for Privacy-preserving Access Control in Cloud-Fog Computing Systems

Yili Jiang*†, Kuan Zhang*, Yi Qian*, and Liang Zhou†
* Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, USA
† College of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, China

2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall) | 978-1-7281-9484-4/20/$31.00 ©2020 IEEE | DOI: 10.1109/VTC2020-Fall49728.2020.9348516

***Abstract*—The cloud-based Internet-of-Things (IoT) has been applied to support ubiquitous data collection and centralized data processing among various applications. Equipped with powerful resources, a semi-trusted cloud is able to deduce private information by launching inference attack. Homomorphic Encryption (HE) has been proposed as an effective way to preserve privacy from inference attack while allowing certain computation over ciphertext. However, HE leads to longer latency due to additional communication and computation overheads. In this paper, we propose an optimization framework in privacy-preserving access control under cloud-fog computing systems. The optimization goal is to maximize the average user satisfaction in the system, where cost and latency serve as key metrics measuring user satisfaction. Due to the NP-hardness of the formulated problem, we propose a low-complexity suboptimal algorithm to solve it, where the access offloading decision making, user cooperation, and resource allocation are considered. Simulation results are presented to show the advantages of our proposed algorithm in terms of the average USI (User Satisfaction Index) and the number of users with zero USI.**

***Index Terms*—privacy-preserving, user satisfaction, access control, cloud-fog.**

I. Introduction

Internet-of-Things (IoT) becomes a key enabler for a variety of applications such as smart home, smart grid, e-health, smart transportation, and so on [1]–[4]. Benefiting from IoT, various devices are connected for data collection and information sharing. Meanwhile, cloud computing is a platform which provides powerful storage and computation resources. By integrating the cloud into IoT, the cloud-based IoT has shown significant potential in centralized data processing and data storage. However, a cloud may derive private information by launching inference attack. Inference attack can deduce sensitive information by analyzing the access behaviors of data requestors. For instance, in e-health applications [5], Alice outsources her encrypted health data to a cloud. If the cloud notices that a doctor from the department of plastic surgery requests the data of Alice for several times in the past two weeks, then it may deduce that Alice had plastic surgery recently. Currently, Homomorphic Encryption (HE) has been applied as an effective way to resist privacy leakage from inference attack in different scenarios [6]–[8]. However, HE results in additional computation and communication overheads, thus causing longer latency to data requestors [9] [10]. Fog computing is proposed to reduce the latency caused by a remote cloud [11]. As a platform located between the devices and the cloud, fog computing is expected to provide storage, computation and communication resources closer to the devices, thus processing latency-sensitive data at the network edge instead of the remote cloud.

However, The cloud-fog computing systems are mainly applied for latency-sensitive task processing [12]–[14]. Most of existing work focuses on task offloading and resource allocation issues. In [12], user fairness is guaranteed by optimizing the task offloading decisions and resource allocation. The authors in [13] improved system performance via jointly managing computation and communication resources. However, data computing in access control is totally different from that in task processing, preventing from applying existing results of task processing into access control directly. Because in task processing, data is uploaded from device to fog or cloud. In access control, however, data is stored in fog and cloud. The distribution of data storage can have a great impact on the access offloading decision.

To deploy the benefits of the cloud-fog computing into HE-based access control systems to achieve lower latency for data requestors, the above challenges must be considered. Different from task offloading, the access offloading in this work is defined as decision making about where a data user can access data. Particularly, an access offloading decision is either a fog node or cloud. To this end, we propose an optimization framework for the privacy-preserving access control in cloud-fog computing systems. The optimization framework combines access offloading and resource allocation to satisfy data users. The main contributions of this paper are summarized as follows.

- Different from existing research work in task processing, an optimization framework is proposed to maximize the average user satisfaction in HE-based access control.
- The optimization problem is formulated as mixed-integer nonlinear programming (MINLP). In the formulation, latency and cost are considered to measure user satisfaction.
- A low complexity algorithm (AUR) is proposed to solve the formulated NP-hard problem. This algorithm is composed of **a**ccess offloading decision making, **u**ser cooperation strategy and **r**esource allocation.

The rest of this paper is organized as follows. In Section II, the system model is provided. In Section III, the formulated optimization problem is described. After that, AUR is proposed in Section IV. Simulation results are analyzed in Section V and conclusions are provided in Section VI.

978-1-7281-9484-4/20/$31.00 ©2020 IEEE

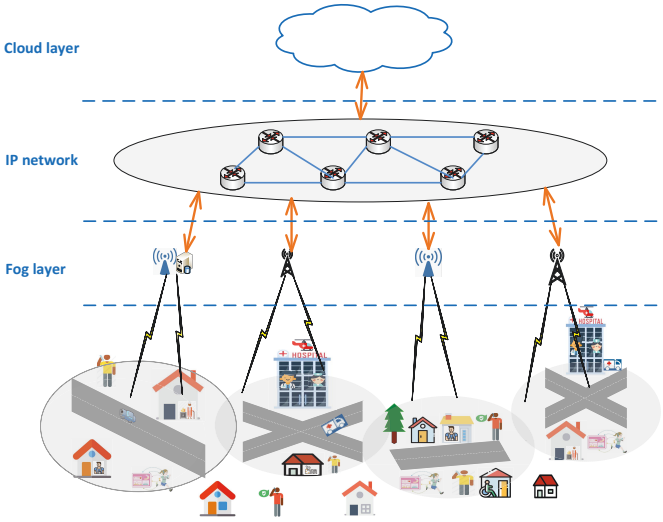

Fig. 1. System Model

II. System Model

A. System Overview

In this paper, we consider a cloud-fog system that supports privacy-preserving access control. As shown in Fig. 1, the system model contains three types of entities: data users, fog nodes and cloud.

- Data users refer to the data requestors in this work. Data users send data requests to their local fog nodes and access data from local fog nodes or cloud.
- Every fog node is responsible for a local area that consists of a certain number of data users. When a fog node receives any request from the data user, it makes access offloading decision to determine whether the data user accesses data from the local fog node or cloud. A fog node only stores part of data in the system due to its limited storage and computation resources.
- One centralized cloud is considered in this work. Unlike a fog node, the cloud stores all data in the system, profiting from its powerful resources.

Cloud and fog nodes are assumed to be semi-trusted. It means that they honestly follow access control policy as requested, but they may be curious to deduce additional information by launching inference attack. In this work, we apply HE in [5] to prevent privacy leakage from semi-trusted cloud and fog nodes.

B. User Satisfaction

As discussed in the previous section, data users may have sensitive latency requirements. To achieve lower latency, more resources are expected. However, more resources come with higher cost. Data users need to consider the balance between latency and cost. Therefore, these two parameters are used to measure user satisfaction in this work. We define the user satisfaction index (USI) of user n as Equation (1).

$$USI_n = 1 - W_1 \times \frac{C_n}{CE_n} - W_2 \times \frac{T_n}{TE_n}, \tag{1}$$

where W_1 and W_2 are weights of cost and latency. CE_n and TE_n are the cost and latency constraints given by user n. C_n and T_n are actual cost and latency that user n achieves.

When implementing HE to prevent from inference attack, additional computation is required to preserve the privacy of role attributes and access policies. For data sets with different sizes, the required computation could be different. The relationship between the necessary computation resource and data size is represented as follows [5].

$$CR_n = \alpha \times D_n. \tag{2}$$

The actual cost of user n denoted as C_n, is defined as follows.

$$C_n = \beta 1 \times \mu_n + \beta 2 \times \lambda_n, \tag{3}$$

where μ_n is the assigned computation capability of user n and λ_n is the assigned bandwidth of user n. $\beta 1$ and $\beta 2$ are the unit costs of computation resource and communication resource respectively.

The actual latency of user n, which is denoted as T_n, includes communication latency and computation latency. It is defined as follows [12][13].

$$T_n = \frac{D_n}{R_n} + \frac{CR_n}{\mu_n}, \tag{4}$$

where $R_n = \lambda_n \times \log_2(1 + SNR)$.

All notations used in this paper are listed in Table I.

III. Problem Formulation

We consider a set of data users $\mathcal{U} = \{u_1, u_2, ..., u_N\}$. Each data user has his/her latency and cost constraints. In fog layer, a set of fog nodes, denoted as $\mathcal{F} = \{f_1, f_2, ..., f_M\}$, is considered. We assume that the computation resource of cloud is unlimited while the computation resource of a fog node is limited. In particular, the communication resource considered in this work is bandwidth, and the computation resource refers to CPU cycle rate. Thus the computation time in cloud can be negligible. Similarly, fog nodes are close to data users, making the communication time between data users and fog nodes are negligible. In other words, if a data user accesses data from cloud, the actual latency is dominated by the communication time. If a data user accesses data from a fog node, the actual latency is dominated by computation time. Note that a data user can only access data from either cloud or a fog node. We aim to maximize the average USI among all data users while meeting their cost and latency constraints. An optimization problem that combines the access offloading

TABLE I
NOTATION DEFINITIONS

Variable	Definition
N	Number of users in the user set
M	Number of fog nodes in the fog nodes set
BW	Total bandwidth in the system
CF_m	Computation capability of fog m
CE_n	Cost constraint of user n
TE_n	latency constraint of user n
C_n	Actual cost of user n
T_n	Actual latency of user n
D_n	Data size required by user n (*bits*)
λ_n	Assigned bandwidth of user n
μ_n	Assigned computation capability of user n (*cycles/second*)
$\rho_{n,s}$	$\rho_{n,0}=1$, if user n accesses data from cloud; $\rho_{n,0}=0$, otherwise. ($s=0$) $\rho_{n,m}=1$, if user n accesses data from fog m; $\rho_{n,m}=0$, otherwise. ($s=m\in[1,M]$)
$\mathcal{CF}$	Set of computation capability of fogs
$\mathcal{CE}$	Set of expected cost of users
$\mathcal{TE}$	Set of expected latency of users
$\mathcal{D}$	Set of data size required by users
$\boldsymbol{\rho}$	Set of data access status of users
$\boldsymbol{\lambda'}$	Set of temporary assigned bandwidth of users
$\boldsymbol{\mu'}$	Set of temporary assigned computation resource of users
$\mathcal{C}'$	Set of bandwidth cost of users
$\mathcal{C}''$	Set of computation resource cost of users
$\mathcal{CF}'$	Set of available computation capability of fogs
BW'	Available bandwidth in the system
$\mathcal{P}$	Set of probability that users change strategy
$\boldsymbol{\lambda}$	Set of final assigned bandwidth of users
$\boldsymbol{\mu}$	Set of final assigned computation resource of users

decision and resource allocation is formulated as follows.

$$\max_{\rho_{n,s},\mu_n,\lambda_n} \sum_{n=1}^{N} \frac{USI_n}{N} \tag{6}$$

$$\text{s.t.} \quad \rho_{n,s} \in \{0,1\}, \tag{7}$$

$$\sum_{s=0}^{M} \rho_{n,s} = 1, \quad \forall n \in [1,N], \tag{8}$$

$$\sum_{n=1}^{N} \rho_{n,0} \times \lambda_n \le BW, \tag{9}$$

$$\sum_{n=1}^{N} \rho_{n,m} \times \mu_n \le CF_m, \quad \forall m \in [1,M], \tag{10}$$

$$T_n \le TE_n, \quad \forall n \in [1,N], \tag{11}$$

$$C_n \le CE_n, \quad \forall n \in [1,N]. \tag{12}$$

(7) and (8) are the constraints on the access offloading decision, indicating that each data user can only access data from either cloud or a fog node. (9) represents that the allocated communication resource cannot exceed the total communication capacity of the cloud. (10) is the computation resource constraint of each fog node. (11) and (12) indicate to satisfy the latency and cost requirements of data users.

Remark 1: The formulated problem is mixed-integer nonlinear programming, which is generally NP-hard.

IV. PROPOSED ALGORITHM

As the formulated problem is NP-hard, it is infeasible to find an optimal solution efficiently. We are motivated to design an algorithm to find a suboptimal solution within limited time.

Algorithm 1 Access Offloading Decision Making

Input: $\mathcal{U}$, BW, $\mathcal{CF}$, $\mathcal{CE}$, $\mathcal{TE}$, $\mathcal{D}$
Output: $\boldsymbol{\rho}$, $\boldsymbol{\lambda'}$, $\boldsymbol{\mu'}$
Initialization: $\boldsymbol{\rho}=\mathbf{0}$, $\boldsymbol{\lambda'}=\mathbf{0}$, $\boldsymbol{\mu'}=\mathbf{0}$, $\mathcal{C}'=\mathbf{0}$, $\mathcal{C}''=\mathbf{0}$, $\mathcal{CF}'=\mathcal{CF}$, $BW'=BW$
Functions: (1) - (4)

```
for all u_n ∈ U do
    Compute λ'_n and μ'_n based on TE_n, (2), (4) and (5);
    Compute C'_n and C''_n based on (3);
    if C'_n <= C''_n and BW' then
        ρ_{n,0} = 1;
        BW' = BW' − λ'_n;
    else if C'_n >= C''_n and CF'_m then
        ρ_{n,m} = 1;
        CF'_m = CF'_m − μ'_n;
    else
        Find a user to change his/her strategy base on Algorithm2;
        Change the strategy of u_j;
        if C'_n <= C''_n then
            ρ_{n,0} = 1;
            BW' = BW' − λ'_n;
        else
            ρ_{n,m} = 1;
            CF'_m = CF'_m − μ'_n;
        end if
    end if
end for
```

The proposed AUR consists of three parts, i.e., **a**ccess offloading decision making, **u**ser cooperation strategy, and **r**esource allocation. Algorithm 1 describes the strategy of access offloading decision making in details. Firstly, a data user is assigned to either cloud or local fog node for data access by making constraint (11) or (12) as equality. For example, by making constraint (11) as equality, we can calculate the necessary resources for a data user and further calculate the minimal communication and computation cost. If the communication cost is lower than the computation cost, the data user then accesses data from cloud. Otherwise, the data user accesses data from the local fog node. Note that it is possible for a data user to be assigned to a resource-limited server. For example, data user u_n is assigned to the cloud to access data, while the communication resource of cloud is not sufficient to satisfy constraint (11). Then the User Cooperation Strategy (UCS) is launched to find the best candidate among data users who have been assigned to the cloud to access data. The candidate changes his/her access decision and access data from his/her local fog node, leaving more communication resource to u_1.

The detailed steps of UCS are summarized in Algorithm 2. All data users assigned to the cloud need to be evaluated whether they can be candidates. A parameter p_i which is defined as the ratio of u_i's required computation resource to the available computation resource of u_1's local fog node. If $p_i < 1$, u_i becomes a candidate. Among all candidates, the one with minimal ratio value is the best candidate. Note that it is possible we cannot find any candidate in this step, then we set the USI of this user as zero directly. It means that this user is not served.

Algorithm 2 User Cooperation Strategy (UCS)

Input: $\boldsymbol{\rho}$, $\boldsymbol{\lambda'}$, $\boldsymbol{\mu'}$, $\mathcal{C}'$, $\mathcal{C}''$, $\mathcal{CF}'$, BW', n
Output: j (index of user who needs to change strategy)
Initialization: $\boldsymbol{P} = \mathbf{0}$

if $C'_n <= C''_n$ **then**
  for all $u_i, i \in [1, n-1]$ **do**
    if $\rho_{i,0}$ **then**
      $p_i = \frac{\mu'_i}{\sum_{m=1}^{M} CF'_m \times \rho_{i,m}}$;
    end if
  end for
else
  for all $u_i, i \in [1, n-1]$ **do**
    if $\rho_{i,0} == 0$ **then**
      $p_i = \frac{\lambda'_i}{BW'}$;
    end if
  end for
end if
Find $\min p_i \in \boldsymbol{P}$;
return $j = i$

After that, the strategy of resource allocation is considered. As the goal is to maximize user satisfaction, we need to consider the trade-off between cost and latency. Particularly, a lower latency responds with a higher cost because more resources are required. In Algorithm 1, data users are assigned with minimal resources which can satisfy constraints (11) and (12). In this step, we try to find the balance between cost and latency by giving extra resources to data users, thus achieving the highest USI for each data user. The detailed description of resource allocation strategy can be found in Algorithm 3. For example, when a data user is assigned to the cloud for accessing data, we can easily derive the balance of cost and latency for this data user based on his/her cost and latency constraints, and then we can calculate how much bandwidth this data user requires to achieve the highest USI, and assign required bandwidth to this user.

V. Performance Evaluations

In this section, we consider a network with one cloud and three fog nodes. The number of data users under each fog node varies within [20, 300]. Different from [12][13], we choose the number of data users per fog instead of the number of users in system as a key parameter, because it presents the user density more accurately. The bandwidth of the cloud is set as

Algorithm 3 Resource Allocation

Input: $\boldsymbol{\rho}$, $\boldsymbol{\lambda'}$, $\boldsymbol{\mu'}$, $\mathcal{CF}'$, BW', BW, $\mathcal{CF}$
Output: $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$
Initialization: $\triangle\boldsymbol{\lambda} = \mathbf{0}$, $\triangle\boldsymbol{\mu} = \mathbf{0}$

if BW' **then**
  for all $u_n \in \mathcal{U}$ **do**
    if $\rho_{n,0}$ **then**
      Compute $\triangle\lambda_n$ to achieve $\max Q_n$;
    else
      $\triangle\lambda_n = \infty$
    end if
  end for
  Rank $\triangle\lambda_n$ from low to high;
  while BW' **do**
    assign $\min \triangle\lambda_x \in \triangle\boldsymbol{\lambda}$ to u_x;
  end while
end if
for all $m \in [1, M]$ **do**
  if CF'_m **then**
    for all $u_n \in \mathcal{U}$ **do**
      if $\rho_{n,m}$ **then**
        Compute $\triangle\mu_n$ to achieve $\max Q_n$;
      else
        $\triangle\mu_n = \infty$
      end if
    end for
    Rank $\triangle\mu_n$ from low to high;
    while CF'_m **do**
      assign $\min \triangle\mu_y \in \triangle\boldsymbol{\mu}$ to u_y;
    end while
  end if
end for
Update$\boldsymbol{\lambda'}$, $\boldsymbol{\mu'}$
return $\boldsymbol{\lambda} = \boldsymbol{\lambda'}$, $\boldsymbol{\mu} = \boldsymbol{\mu'}$

BW = 5 MHz. The computation resource of fog nodes is set as a random distribution from 50 to 60 Gcycles/second. The data size is randomly distributed from 0.5 to 10 Mbits, and latency constraint for each data user is randomly distributed from 0.3 to 2 seconds. We compare the proposed algorithm with the random method [13], which refers to a random access offloading decision making. In [13], if a data user is randomly assigned to a resource-limited server, the data server cannot be served due to insufficient resource. Thus the USI of this data user is zero. In addition, we evaluate the AUR without UCS. Our simulation results show the performance under the comparison of three methods (the random method, the proposed AUR, and the AUR without UCS). Note that each point in the following figures is based on the average values of 1000 simulation runs.

Fig. 2 evaluates the performance of average USI. We take W_1 = 0.1 as an example and increase the number of users per fog from 20 to 300 by the step of 10 to show the performance of average USI. Apparently, for AUR and the AUR without

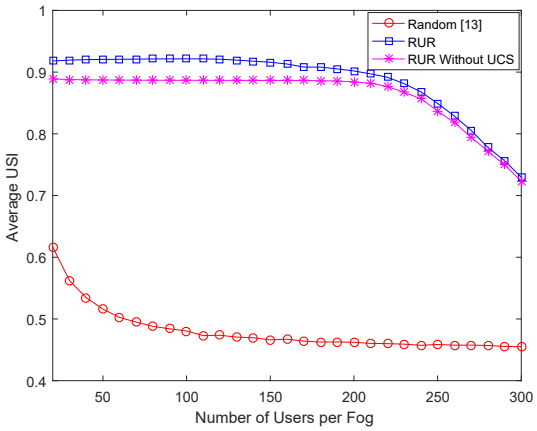

Fig. 2. The average USI vs. the number of users per fog

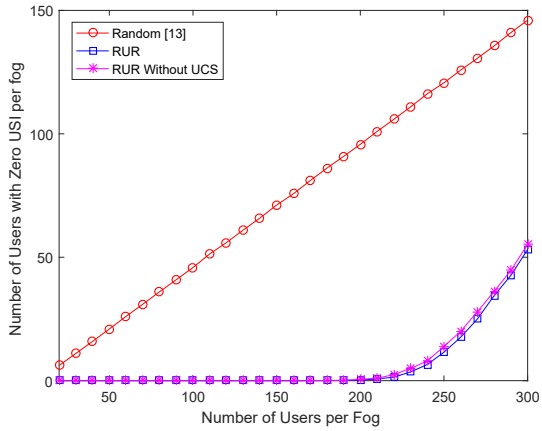

Fig. 3. The number of unserved users per fog vs. the number of users per fog

UCS, the average USI is almost steady and starts to decrease when the number of users per fog is more than 200. This is because with the increase of number of users, more users are not served due to limited resource. Then the number of users with zero USI rises, reducing the average USI. AUR performs better than the AUR without UCS when number of users per fog is less than 200. In other words, UCS can efficiently improve the average USI in AUR. However, the gap is narrowed down when the number of users per fog is greater than 200. For the random method, the average USI goes down greatly when the number of users per fog is less than 100 and keeps decreasing slightly after that.

Fig. 3 shows the evaluation about the number of users with zero USI. Intuitively, for AUR and the AUR without UCS, all users can be served when the number of users is less than 200. This is reasonable since resource is sufficient. However, when more users require resource, it is possible that some users cannot be served due to limited resources. For random method, it has users with zero USI even when the number of users per fog is very small due to the randomness of access offloading decision making. Therefore, the data showed in Fig. 3 also explains the result of Fig. 2.

VI. Conclusion and Future Work

In this paper, we have proposed an optimization framework for the privacy-preserving access control in cloud-fog computing systems. To achieve maximal user satisfaction, an optimization problem has been formulated. Due to the NP-hardness of the formulated problem, we have designed AUR, which jointly involves access offloading decision making, user cooperation, and resource allocation, to find a suboptimal solution with low time complexity. The simulation results have demonstrated that our proposed algorithm can achieve a higher average USI and provide service to more users. In the future work, we will further improve the average USI by employing the cooperation between fog nodes.

References

[1] P. Verma and S. K. Sood, "Fog Assisted-IoT Enabled Patient Health Monitoring in Smart Homes," *in IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1789-1796, June 2018.

[2] E. Spanò, L. Niccolini, S. D. Pascoli and G. Iannacconeluca, "Last-Meter Smart Grid Embedded in an Internet-of-Things Platform," *in IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 468-476, Jan. 2015.

[3] G. Yang, M. Jiang, W. Ouyang, G. Ji, H. Xie, A. M. Rahmani, P. Liljeberg and H. Tenhunen, "IoT-Based Remote Pain Monitoring System: From Device to Cloud Platform," *in IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 6, pp. 1711-1719, Nov. 2018.

[4] H. A. Khattak, H. Farman, B. Jan and I. U. Din, "Toward Integrating Vehicular Clouds with IoT for Smart City Services," *in IEEE Network*, vol. 33, no. 2, pp. 65-71, March/April 2019.

[5] W. Zhang, Y. Lin, J. Wu and T. Zhou, "Inference Attack-Resistant E-Healthcare Cloud System with Fine-Grained Access Control," *in IEEE Transactions on Services Computing*, Jan. 2018.

[6] K. Zhang, X. Liang, J. Ni, K. Yang and X. (Sherman) Shen, "Exploiting Social Network to Enhance Human-to-Human Infection Analysis without Privacy Leakage," *in IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 607-620, 1 July-Aug. 2018.

[7] A. Abdallah and X. S. Shen, "A Lightweight Lattice-Based Homomorphic Privacy-Preserving Data Aggregation Scheme for Smart Grid," *in IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 396-405, Jan. 2018.

[8] B. Jin, D. Jiang, J. Xiong, L. Chen and Q. Li, "D2D Data Privacy Protection Mechanism Based on Reliability and Homomorphic Encryption," *in IEEE Access*, vol. 6, pp. 51140-51150, 2018.

[9] W. Wang, Y. Hu, L. Chen, X. Huang and B. Sunar, "Exploring the Feasibility of Fully Homomorphic Encryption," *in IEEE Transactions on Computers*, vol. 64, no. 3, pp. 698-706, March 2015.

[10] A. Chatterjee and I. Sengupta, "Translating Algorithms to Handle Fully Homomorphic Encrypted Data on the Cloud," *in IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 287-300, 1 Jan.-March 2018.

[11] Cisco System, "FogComputing and the Internet of Things: Extend the Cloud to Where the Things Are," 2015, white paper. [Online]. Available: https://www.cisco.com/c/dam/en˙us/solutions/trends/iot/docs/computing-overview.pdf

[12] J. Du, L. Zhao, J. Feng and X. Chu, "Computation Offloading and Resource Allocation in Mixed Fog/Cloud Computing Systems With Min-Max Fairness Guarantee," *in IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1594-1608, April 2018.

[13] Y. Gu, Z. Chang, M. Pan, L. Song and Z. Han, "Joint Radio and Computational Resource Allocation in IoT Fog Computing," *in IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7475-7484, Aug. 2018.

[14] A. Boukerche, S. Guan and R. E. De Grande, "A Task-Centric Mobile Cloud-Based System to Enable Energy-Aware Efficient Offloading," *in IEEE Transactions on Sustainable Computing*, vol. 3, no. 4, pp. 248-261, 1 Oct.-Dec. 2018.