Secure and Sustainable Load Balancing of Edge Datacenters in Fog Computing

Deepak Puthal[*], Mohammad S. Obaidat, *Fellow, IEEE*[†], Priyadarsi Nanda[*], Mukesh Prasad[*], Saraju P. Mohanty[‡], and Albert Y. Zomaya, *Fellow, IEEE*[¶]

[*]Faculty of Engineering and IT, University of Technology Sydney, NSW, Australia
[†]King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan
[‡]Department of Computer Science and Engineering, University of North Texas, TX, USA
[¶]School of Information Technologies, The University of Sydney, NSW, Australia
Corresponding Email: deepak.puthal@gmail.com, msobaidat@gmail.com, albert.zomaya@sydney.edu.au

***Abstract*—Fog computing is a recent research trend to bring cloud computing services to network edges. Edge datacenters (EDCs) are deployed to decrease the latency and networks congestion by processing data streams and user requests in near real-time. The EDCs deployment is distributed in nature and positioned between cloud datacenter and data sources. Load balancing is the process of redistributing the work load among EDCs to improve both resource utilization and job response time. Load balancing also avoids a situation where some EDCs are heavily loaded while others in idle state or doing small data processing. In such scenarios, load balancing between the EDCs plays a vital role for user response and real-time event detection. As the EDCs are deployed in the unattended environment, secure authentication of EDCs is an important issue to address before performing load balancing. This paper proposes a novel load balancing technique to authenticate the EDCs and find out less loaded EDC for task allocation. The proposed load balancing technique is more efficient than other existing approaches in finding less loaded EDC for task allocation. The proposed approach not only improves efficiency of load balancing, it also strengthens the security by authenticating the destination EDCs.**

***Index Terms*—Fog Computing, Edge Computing, EDC, Cloud, Security, Load Balancing.**

I. Introduction

Fog computing exhibits some of the overlapping features of cloud with additional attributes such as location awareness and edge datacenter (EDC) deployment. A large number of EDCs are geographically distributed to offer mobile, low latency data transparency over real-time request and responses [1]. Cloud computing is popular to scalable computation and processing of large amount of data (termed as bigdata). This is also popular for storage, and provisioning of resources according to user requirements. In recent days, fog computing has been proposed to migrate the cloud resources to the EDCs, where EDCs are deployed across network edges [7]. There are several fog computing architectures, listed with edge deployment. Figure 1 presents a block diagram of the three layers in fog computing architecture. The bottom layer includes several terminal devices such as wireless sensor nodes and smart devices, where these devices transmit data to the upper layers [14], [15]. In the second layer, the fog contains highly intelligent devices, such as routers, switches and gateways. In some architecture, middle layer (Edge layer) divided into two parts such as edge device and edge datacenter, but most of the fog computing architectures combine these two to form a single layer. The third and topmost layer tends to be the cloud datacenter comprising several high-end servers. Cloud datacenters with user response facilities. The combination of these three layers defined as fog computing architecture and the comprehensive architecture with different module are presented in Figure 2.

With the great advancements in computing environment and the availability of EDCs services in fog computing, the problem of load balancing of EDCs has gained high attention and importance. There are numerous research works that have been conducted to solve the load balancing problem. However, none of them adequately addressed EDC authentication issue. As,

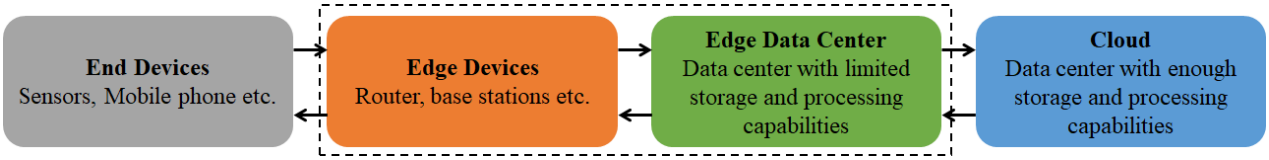


Figure 1. Three-layer block diagram of the fog computing architecture with inter layer da flow.

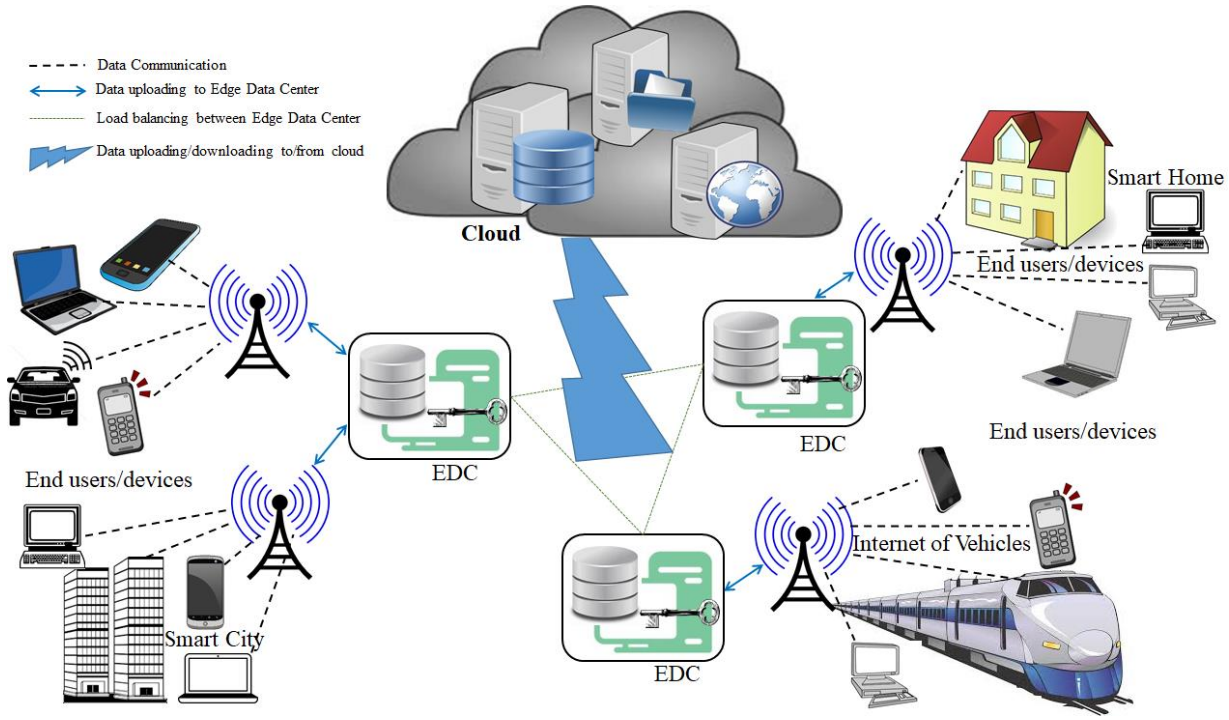


Figure 2. A comprehensive architecture of the fog computing with the individual components and EDCs inter communication.

the EDCs are deployed in the network edges in an unattended scenario, authentication of EDC become a key factor before load balancing. All the EDCs are deployed in a distributed environment, so load balancing should work in a distributed scenario. Load balancing in distributed environments are divided into two main approaches i.e. static load balancing and dynamic load balancing [2].

In the static load balancing, load balancing is achieved by providing a set of tasks to specific EDCs so that the performance function is minimized. This load balancing is done with either deterministic or a probabilistic means. In a deterministic balancing technique, EDC-I allocates the over loaded tasks to the EDC-J all the time. Whereas in deterministic balancing technique, EDC-I allocates the overloaded tasks to EDC-K with probability *x* and to EDC-L with probability *y*. The major drawback of the static load balancing is that it does not consider the status of the destination EDC while making decision of load balancing. In the dynamic load balancing, the current load status of the individual EDCs is considered to decide the destination EDC. As a result, tasks are assigned dynamically from an over loaded EDC to under loaded EDC for efficient computing. Even though dynamic approach is much difficult to achieve, it always gives a better solution towards sustainable load balancing. For this reason, the paper considers a dynamic load balancing technique to design the proposed solution.

There are several authentication methods available for the networks systems, however there is no authentication solution for the EDC. To the best of our knowledge, this paper proposes a secure authentication method to select the authenticated EDC for the load balancing for the first time. The main contributions of the proposed approach are summarized below:

- The proposed approach presents an adaptive EDC authentication technique with the help of a centralize cloud datacenter. This authentication is initiated by the cloud and then all EDCs authenticate each other by following cloud credentials.
- The proposed approach brings a sustainable and dynamic load balancing technique by considering the load of the destination EDCs. This load information is shared during the authentication process, so that individual EDCs do not need additional communication to get the load information from others.
- Finally, the proposed approach combines both the authentication and load balancing technique to apply in the EDCs. The proposed approach also evaluates the performance by validating the efficiency and scalability.

The rest of this paper is organized as follows. Section 2 gives the related works. Section 3 describes the proposed solution for the secure and sustainable load balancing of the EDCs. Section 4 presents the formal security analysis and verification of our model. Section 5 evaluates the performance and efficiency of the proposed solution through extensive experimentations. Section 6 covers the conclusions and potential future directions.

II. Related Works

Starting from IoT smart sensing device to mobile users, which move randomly and tend to offload tasks to their nearest EDC [1]. Hence, the load states of EDCs in various locations differ greatly. Furthermore, unbalanced problem emerges, as some EDCs in the region could be overloaded while some other EDCs are in idle state [4]. There exist several previous works that proposed different methods to address load balancing issues.

A. Load Balancing

By formulating the load balancing problem in EDCs as an optimization problem, Jia et al. [5] proposed scalable algorithm to find a redirection of tasks between a given set of EDCs in a network thus minimizing the maximum of average response time. Willebeek-LeMair and Reeves [6] proposed a basic dynamic load balancing for distributed systems in 1993. Followed by several researchers who have contributed to make it efficient for different scenarios and applications. Tong et al. [7] proposed a novel technique to handle the peak load and satisfy the requirements of remote program execution. Other authors have deployed the cloud servers in network edges to design edge-computing architecture and proposed a workload placement algorithm to maintain the load balancing efficiently. The geographical load balancing is achieved by routing workloads dynamically to reduce overall energy consumption [8]. Zhang et al. [8] proposed an algorithm to solve the challenging load balancing problem optimally and efficiently by discovering the entire design space of strategic bidding.

B. Security Issues

The very basic study on security issues of EDCs with cyber threats has been classified in [9], which discussed the need of security in EDC deployment. There are several existing works that proposed different methods to address the authentication problem in network scenario. A cloud-centric multi-level authentication scheme was proposed in [10] to addresses scalability, time constraints, and effectiveness of the scheme. Butun et al. [10] have proposed this authentication technique for connected devices in Internet of Things and public safety. By focusing on healthcare technology, He and Zeadally [11] proposed an efficient authentication technique for body area networks after discussing the overall system architecture with associated security requirements. He et al. [12] have proposed an anonymous authentication method for same wireless body area network and validated with proofs. In current cyber threats scenario, the best way is to protect the system identification by developing a security perimeter [13]. This remains a challenging task for the EDC, as EDCs are deployed in an open access networks. There is a need for a new authentication scheme for EDC load balancing. Therefore, this paper proposes a new solution after the EDC load balancing.

III. Proposed Method for Load Balancing

Based on the current literature survey, there is no such architecture to authenticate the edge datacenter before allocating tasks. So, this paper proposes a novel architecture to not only authenticate, but also get the load information of the EDCs before sharing the tasks. The complete procedure of the load balancing technique is described in the following two subsections, which first discuss the secure authentication of the EDCs and then the sustainable load balancing technique.

A. Secure Authentication

Based on the fog computing architecture, all the data are stored and processed at the cloud, where EDCs work as the intermediate datacenters to reduce the latency of the user requests. Cloud is always deployed in the secure environment, so we have considered cloud to initiate the authentication process. Cloud initiates the process to assign initial ID (Ei) associated with the key (Ki) and shared key (Kc) for the individual EDCs during the EDCs deployment. EDCs use trusted modules (such as Trusted Platform Module (TPM)) to store the secret information from the cloud and the rekeying process [3]. After initialization of the EDCs, each individual EDC starts to authenticate the EDCs in the region. This helps in future to avoid the malicious EDCs to participate in load balancing.

Let us assume EDC-I is the edge datacenter, which starts the authentication process. It combines its own ID with the associate key and encrypts using the shared key initiated by the Cloud $\left(E_{K_c}(E_i \parallel K_i)\right)$. EDC-I broadcasts the generated packets by sending to all the EDCs in the region. When other EDCs get the authentication request packet, they decrypt it using the cloud shared key$\left(D_{K_c}(E_i \parallel K_i)\right)$. As the cloud shared key is the same for all the EDCs, they can use the same key to perform the encryption and decryption process. The shared key (Kc) is initiated by the cloud to individual EDCs and all the EDCs trust each other with this shared key. Once

destination EDC (EDC-J) gets the source ID and its associate key, it checks with the cloud to confirm authenticity of the source EDC. Once everything is confirmed by cloud, it keeps a copy of the EDC-I details as an authenticated EDC. Then EDC-J concatenates own ID with the associated key and encrypts it using source associate key $\left(E_{K_i}(E_j \parallel K_j)\right)$. Once EDC-I receives the encrypted packet, it will decrypt it with its own key and followed by sending it to the cloud for verification of EDC-J. The encrypted packet is of the format $\left(E_{K_c}\left(E_i \parallel E_{K_i}(E_j)\right)\right)$, where E_j is encrypted with source EDC-I associate key. This is combined with its own ID to generate encrypted packet using cloud shared key. After receiving the encrypted packet at cloud datacentre, it decrypts it using the shared key and then retrieves the associated key of $E_j(E_j \rightarrow K_j)$ to validate the EDC-J. Once it is validated, the cloud concatenates E_j and the associate key and encrypts it with EDC-I associated key to send it back to EDC-I. After receiving the encrypted packet, EDC-I decrypts it to find the key (K_j') and it compares it with the associated key received from EDC-J. If a match found that $K_j = K_j'$, then EDC-I combines the ID of EDC-I and EDC-J and encrypts it with destination associate key (kj). Once this combined packet is received by the EDC-J, it confirms that both EDC-I and EDC-J are now authenticated to each other for load balancing. The stepwise complete procedure of this authentication process is shown in Figure 3. Individual EDCs generate their public $\left(PuK_{i_{/j\ldots}}\right)$ and private $\left(PrK_{i_{/j\ldots}}\right)$ key pairs and broadcast the public key $\left(PuK_{i_{/j\ldots}}\right)$ to be used for further use by the EDC in load balancing. Source EDC always uses recipient EDC's public key to encrypt the loads before sharing loads. For more details about the use of public and private keys, see next subsection.

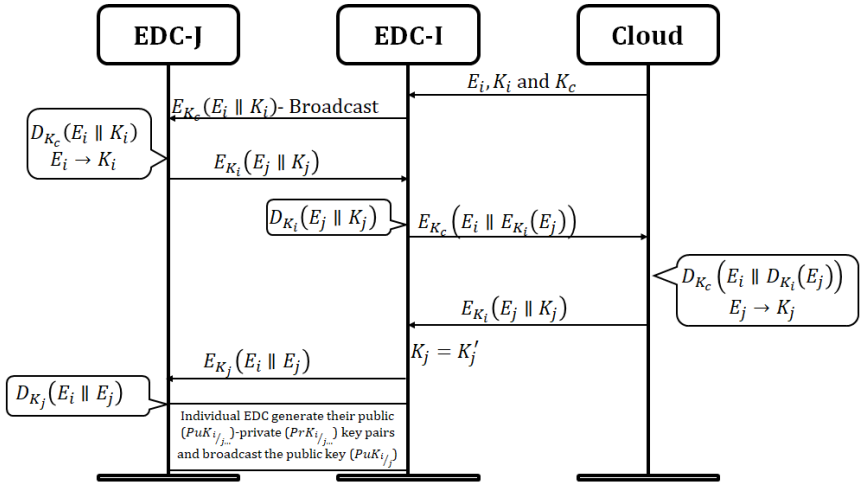


Figure 3. Stepwise information and credential flow for the secure authentication of EDCs.

B. *Sustainable Load Balancing*

This paper follows BFS (Breadth First Search) method to design the proposed load balancing technique. We have used two parameters i.e. m and n to maintain the load of all the EDCs, where m is the current load and n is the maximum capacity to process the tasks. In order to compute the current load statues, we use a parameter p, where $p = {}^{m}/_{n}$. Individual EDCs get load balancing requests from other EDCs to process their tasks.

If EDC-I is over loaded, then EDC-I broadcasts a control packet by sending requests to other EDCs in the region with its own ID and the received load information i.e. (*Ei*, *Li*). The ID of EDC-I is defined as *Ei*, whereas the received load information is defined as *Li*. The neighbor EDC (named as EDC-J) checks the received ID and compares it with its own database. In case of a match, EDC-J then looks for the load information from the control packets, otherwise, it ignores the control packet to avoid the DoS attack.

While checking the EDC-I load information, EDC-J first checks its own load information using value of *p*. If *p* is less than or equal to 0.6, then it moves forward to get the available resources (i.e. *n*-*m*) to process the invited tasks. If the available resource is higher than the required resource to process the invited task, then EDC-J processes the positive response packet to the EDC-I. Otherwise, EDC-J becomes silent without any response. While preparing the response, EDC-J includes its own identity (E_j), associate key and the current free resource of the datacenter (i.e. *p*). Finally, it generates the response packet, encrypts it with the public key of the destination EDC i.e. kpu_i, $\left(E_{Kpu_i}(E_j||K_j||p)\right)$ and sends it to the required destination EDC-I. After receiving the encrypted data packets, EDC-I uses its own private key i.e. Kpr_i to decrypt the data packets $\left(D_{Kpr_i}(E_j||K_j||p)\right)$. After decryption, EDC-I checks the source ID (E_j) and compares

it with its own database to find if there is a match. If a match is found, it extracts the associate key with the ID (k_j'). Also, EDC-I compares the associated key with the received key ($k_j' = k_j$) and in case of a match, it accepts the response otherwise, it rejects it. In a similar way, EDC-I gets several responses from different EDCs in the region. EDC-I also compares the values of neighbor *p* and finds the lowest value to select destination. Finally, EDC-I sends tasks to the authenticated neighbor EDCs to process. The stepwise procedure of the load balancing technique is shown in Algorithm 1.

Algorithm 1. Load Balancing Technique

1. If (EDC-I is overloaded)
2. EDC-I broadcast (E_i, L_i)
3. EDC-J (neighbor EDC) verifies:
4. If (E_i is in database) & ($p \leq 0.6 \& L_i << (n-m)$)
5. Response $E_{Kpu_i}(E_j||K_j||p)$
6. EDC-I perform $D_{Kpr_i}(E_j||K_j||p)$
7. $k_j' \leftarrow E_j$
8. If ($k_j' = k_j$)
9. EDC-I select EDC-J for load balancing.

IV. Security Evaluation

We evaluated the secure authentication model using theoretical analysis and formal verification, which are discussed below.

A. Security proof

Definition (Attack on authentication). An intruder "*Ma*" attacks on authenticity and is capable of monitoring, intercepting, and introducing itself as an authenticated EDC to start load balancing process. The types of possible attacks in this category include impersonation attack and identity-based attacks [3].

Claim: An attacker *Ma* cannot read the secret credentials of EDC to introduce itself as an authenticated EDC to participate in load balancing.

Proof: Following the above definition of attack on authenticity and computational hardness of TPM module (a secure module of EDC), we believe that attacker *Ma* cannot get the secret information for *Ei*, *Ki* and *Kc* initiated by the cloud. All the secure information to perform the authentication process are initiated by the cloud during the EDC deployment. When EDCs start authenticating each other, they use cloud shared key (*Kc*) to encrypt the initial authentication packet $\left(E_{K_c}(EDC_i \parallel K_i)\right)$ followed by individual associate keys of EDCs (*Ki/j*). It is followed by AES based symmetric encryption during the initial authentication. So, the transaction cannot be broken for years [3]. Thus, it is close to impossible to monitor the network thoroughly and get the authentication credentials. During the authentication process, individual EDCs use their secure module to perform the encryption or decryption or to save their keys. Hence, it is nearly impossible to get either process or keys from secure module, from TPM properties. Consequently, we conclude that an attacker *Ma* cannot attack on authenticity during EDC load balancing.

B. Formal security verification

The formal verification of the proposed secure authentication scheme is written using Scyther simulation environment. Scyther uses Security Protocol Description Language (.spdl) to design the security methods and flows. By following Scyther features, we define the role of *Ei* and *Ej*, where *Ei* is the initiator for the authentication (EDC-I) and *Ej* is the destination EDC of authentication (EDC-J). In our verification scenario, *Ei* and *Ej* have all the required security information with them, which are initiated by the cloud. *Ei* starts sending the packets to *Ej* and *Ej* responses with the load information. In the verification scenario, this paper introduces the authentication attack i.e. where an adversary acquires the authentication property of *Ei* and sends the malicious packets to *Ej* to start the load balancing process. The experiment uses 100 runs with 10 intervals to check the possible attacks on authenticity. Apart from these, this paper follows the default properties of Scyther to run the simulation.

While considering the attack models, there are several attacks which can be possible, however in this case, we focus on authentication attacks as discussed before. In authentication attacks, an attacker can observe any EDC communication continuously in order to discover the authentication patterns. We assume that any malicious EDC gets the authenticated EDC's behavior and starts the communication process for load balancing. The proposed solution uses trusted modules (such as TPM) of

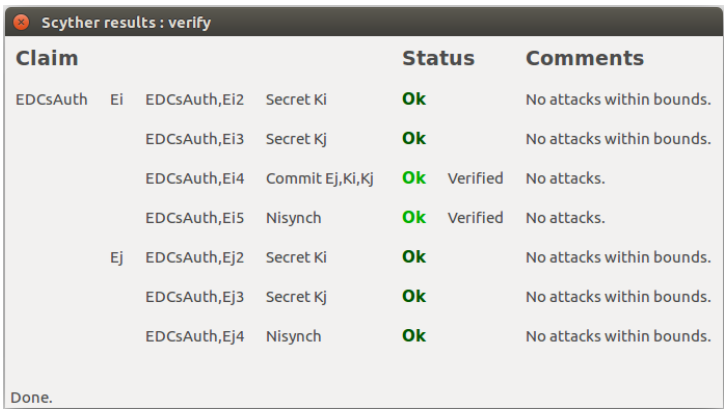
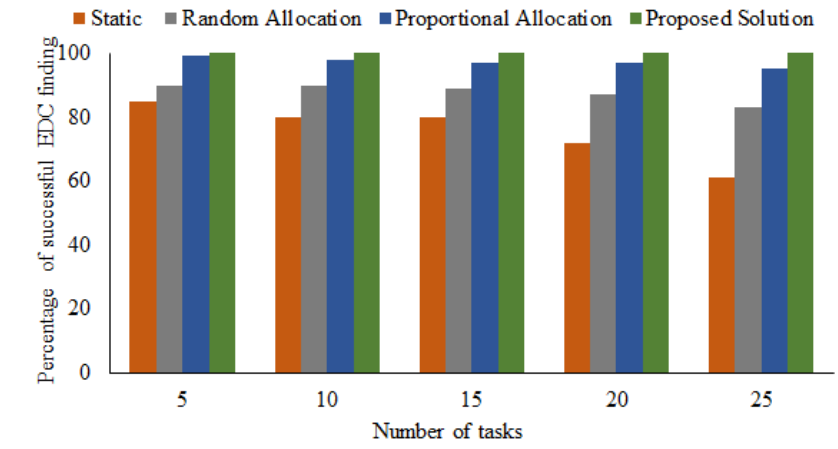
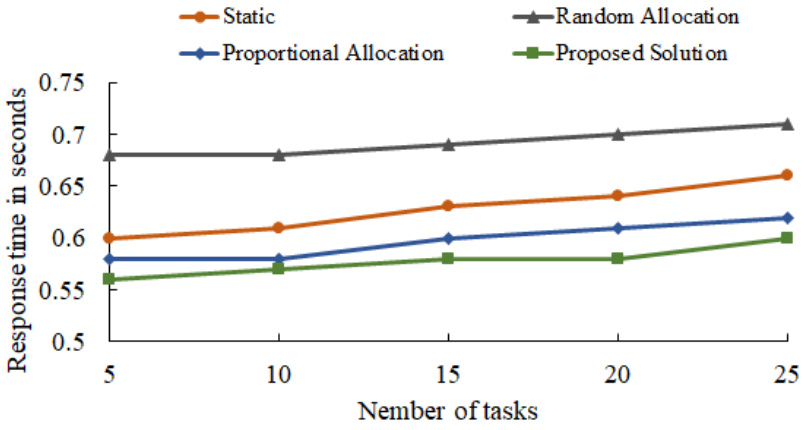


Figure 4. Scyther formal security verification result page.



(a) Percentage of successfully selecting the destination ECS for load balancing.



(b) Response time of destination EDC for load balancing.

Figure 5. Simulation results of proposed scheme with comparison to existing techniques.

the EDC to store the rekeying process and secret keys.

The experiment runs in the Scyther environment for 100 instances with 10 inter as described above. During the whole experimental process, we did not encounter any attack while focusing mostly on authentication attack. The security verification result of Scyther environment is shown in Figure 4, which shows that the proposed security solution is secured against authentication attack.

V. Experiment and Results

The performance of the proposed load balancing solution is evaluated using the Matlab simulation environment. We run the program in a Dell computer with Intel Core i7 processor and 8 GB RAM. All the simulations are executed for 10 times, and average values from the execution are considered to validate the scheme. In our experiments, we initiated ten EDCs for evaluating overall performance of the proposed scheme. We considered task arrival rate λ_i with Poison arrival process. During simulation, we assumed that, EDC-I is already over loaded and hence, required to perform load balancing after getting any extra tasks. EDC-I starts the load balancing process with authentication and gets the destination EDCs load information to find any suitable and less loaded EDCs to allocate the tasks. After querying destination's load information, the task is assigned to the less loaded ones. We also evaluate three benchmark schemes in simulation, i.e., random allocation, proportional allocation and static allocation load balancing techniques. In random allocation, a mobile cloudlet offloads tasks to a randomly picked neighbor. On the contrary, the proportional allocation scheme queries global load information from cloudlet in the neighboring list and selects optimal one to offload a task. However, in static method the task is allocated to a specific destination for all the time.

First, the simulation result gets the performance of the proposed load balancing technique to select the suitable and less loaded EDC for allocation of the tasks. We follow similar simulation setup as described above to simulate the scenario. The simulation results of successfully finding the destination EDC is shown in Figure 5(a). The successful heat ratio is calculated in percentages (%), and we found that the proposed load balancing solution gives 100% success rate to find the most suitable and less loaded EDC. We compared the performance with static, random and proportional allocation. Proportional allocation gives always better result compared to others as it considers destination EDC's load before allocating the task. Whereas, we found that the proposed technique looks better and consistent even with increasing number of tasks. Hence, the proposed solution is secured as well as efficient in selecting destination EDC for load sharing.

Response time of the destination EDC also plays a vital role to improve the efficiency of the overall processing time. We also considered the same simulation

setup as above to compute the response time. In the similar way, we compared the performance of the proposed load balancing solution with static, random, and proportional allocation. The result of response time performance metric is shown in Figure 5 (b). From the result, we found that proportional allocation always provides better performance compared to the other two existing techniques. However, the proposed load balancing performs better than the proportional allocation technique. At the same time, the proposed solution also authenticated the destination EDCs before load balancing. This shows that the proposed load balancing solution has better response time compared to other existing techniques even after the secure authentication process.

From above theoretical and experimental evaluation, we conclude that the proposed load balancing solution is not only sustainable, but also secured. This improves the load balancing performance of EDCs in fog computing environment.

VI. Conclusion

This paper proposes a novel secured and sustainable load balancing solution for EDCs in fog computing environment. The proposed load balancing technique is basically divided into two major parts, where the first part focuses on secure authentication of the EDCs in the region by using cloud initiated credentials, and followed by a sustainable load balancing architecture by getting load information of the destination EDCs. The proposed solution has been evaluated in two different ways both using theoretical analysis and experimental evaluation. From the performance evaluation and comparison results, we conclude that the proposed solution is secured and sustainable by getting destination EDC's load during authentication process. As EDCs are deployed in the open and hostile environment, we proposed security solution to protect against the outsider attacks to authenticate the destination EDCs by avoiding malicious ones.

In the future, we plan to extend our research avenues by proposing lightweight security solution and improve load balancing performance of EDCs in fog computing environments. In addition to this, we are building a real-time testbed to implement the proposed security and load balancing scheme.

References

[1] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K-K R. Choo, and M. Dlodlo. "From Cloud to Fog Computing: A Review and a Conceptual Live VM Migration Framework." *IEEE Access*, Vol. 5, pp. 8284-8300, 2017.

[2] A. Alakeel. "A guide to dynamic load balancing in distributed computer systems." International Journal of Computer Science and Information Security, Vol. 10, no. 6, pp. 153-160, 2010.

[3] D. Puthal, S. Nepal, R. Ranjan, and J. Chen. "DLSeF: a dynamic key-length-based efficient real-time security verification model for big data stream." *ACM Transactions on Embedded Computing Systems*, Vol. 16, no. 2, pp. 51, 2017.

[4] K-K R. Choo, R. Lu, L. Chen, and X. Yi. "A foggy research future: Advances and future opportunities in fog computing research." *Future Generation Computer Systems*, Vol. 78, no. 2, pp. 677-679, 2018.

[5] M. Jia, W. Liang, Z. Xu, and M. Huang, "Cloudlet load balancing in wireless metropolitan area networks," in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pp. 1–9, 2016.

[6] M. Willebeek-LeMair, and A. Reeves. "Strategies for dynamic load balancing on highly parallel computers." *IEEE Transactions on parallel and distributed systems*, Vol. 4, no. 9, pp. 979-993, 1993.

[7] L. Tong, Y. Li, and W. Gao. "A hierarchical edge cloud architecture for mobile computing." In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pp. 1-9, 2016.

[8] Y. Zhang, L. Deng, M. Chen, and P. Wang. "Joint Bidding and Geographical Load Balancing for Datacenters: Is Uncertainty a Blessing or a Curse?" In *Computer Communications, IEEE INFOCOM 2017-The 36th Annual IEEE International Conference on*, pp. 1-9, 2017.

[9] D. Puthal, S. Nepal, R. Ranjan, and J. Chen. "Threats to networking cloud and edge datacenters in the Internet of Things." *IEEE Cloud Computing*, Vol. 3, no. 3, pp.64-71, 2016.

[10] I. Butun, M. Erol-Kantarci, B. Kantarci, and H. Song. "Cloud-centric multi-level authentication as a service for secure public safety device networks." *IEEE Communications Magazine*, Vol. 54, no. 4, pp. 47-53, 2016.

[11] D. He, and S. Zeadally. "Authentication protocol for an ambient assisted living system." *IEEE Communications Magazine*, Vol. 53, no. 1,pp. 71-77, 2015.

[12] D. Puthal, S. Nepal, R. Ranjan, and J. Chen. "A Synchronized Shared Key Generation Method for Maintaining End-to-End Security of Big Data Streams." In Proceedings of *the 50th Hawaii International Conference on System Sciences*, pp. 6011-6020, 2017.

[13] D. Puthal, S. Mohanty, P. Nanda, and U. Choppali. "Building Security Perimeters to Protect Network

Systems against Cyber Threats." *IEEE Consumer Electronics Magazine*, Vol. 6, no. 4, pp. 71-77, 2017.
[14] M. S. Obaidat, and P. Nicopolitidis. "Smart cities and homes: Key enabling technologies." *Morgan Kaufmann*, 2016. ISBN: 978-0-12-803454-5
[15] M. S. Obaidat, and S. Misra. "Principles of wireless sensor networks." *Cambridge University Press*, 2014. ISBN: 978-0-521-19247-7

BIOGRAPHIES

Deepak Puthal (M'17) is a Lecturer (Assistant Professor) in the Faculty of Engineering and IT at University of Technology Sydney (UTS), Australia. His research interests include cyber security, Internet of Things, distributed computing, and Big Data Analytics. He has Ph.D. in Computer and Information Systems from UTS, Australia. He is the winner of IEEE Distinguished Doctoral Dissertation Award for Excellence in STC on Smart Computing for the year 2017.

Mohammad S. Obaidat (F'05) received Ph.D. and M.S. degrees in computer engineering with a minor in computer science from Ohio State University, USA. He is a well-known worldwide academic/scientist/academic. He is currently a Full Professor at King Abdullah II School of Information Technology, University of Jordan. He has published about 55 books, over 55 book chapters and over 700-refereed technical journal and conference articles.

Priyadarsi Nanda (SM'14) earned his Ph.D. degree from the University of Technology Sydney, Australia. He is a senior lecturer in the School of Computing and Communications at University of Technology Sydney, Australia. He has more than 26 years of experience in the area of cybersecurity, Internet of Things security, networks quality of service, assisted health care using sensor networks, and wireless sensor networks. He has published more than 70 refereed re - search articles.

Mukesh Prasad (M'16) has received the M.S. degree from the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India, in 2009 and PhD with the Department of Computer Science, National Chiao Tung University (NCTU) in 2015. I started my academic career as Lecturer with University of Technology Sydney in March 2017. I have made substantial contributions to the field of machine learning and image processing with 41 research articles.

Saraju P. Mohanty (SM'08) is a professor in the Department of Computer Science and Engineering at the University of North Texas, Denton. He is an inventor of four U.S. patents and is an author of 220 peer-reviewed research articles and three books. He is currently the Editor-in-Chief of IEEE Consumer Electronics Magazine.

Albert Y. Zomaya (F'04) is the Chair Professor of High Performance Computing & Networking and Australian Research Council Professorial Fellow in the School of Information Technologies, Sydney University. He is also the Director of the Centre for Distributed and High Performance Computing. He served as the Editor in Chief of the IEEE Transactions on Computers (2011-2014) and was elected recently as a Founding Editor-in-Chief for the newly established IEEE Transactions on Sustainable Computing.