A secure sharing control framework supporting elastic mobile cloud computing

Aws Hamed Hamad[1], Adnan Yousif Dawod[2], Mohammed Fakhrulddin Abdulqader[3], Israa Al_Barazanchi[3,4], Hassan Muwafaq Gheni[5]

[1]Ministry of Higher Education and Scientific Research, Baghdad, Iraq
[2]Department of Computer Science and Information Technology, College of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq
[3]Computer Engineering Techniques Department, Baghdad College of Economic Sciences University, Baghdad, Iraq
[4]College of Engineering, University of Warith Al-Anbiyaa, Karbala, Iraq
[5]Computer Techniques Engineering Department, Al-Mustaqbal University College, Hillah, Iraq

Article Info

Article history:

Received Jun 3, 2022
Revised Oct 1, 2022
Accepted Oct 29, 2022

Keywords:

Mobile cloud computing
Module allocation
Risk control
Security framework
Security threats

ABSTRACT

In elastic mobile cloud computing (EMCC), mobile devices migrate some computing tasks to the cloud for execution according to current needs and seamlessly and transparently use cloud resources to enhance their functions. First, based on the summary of existing EMCC schemes, a generic EMCC framework is abstracted; it is pointed out that the migration of sensitive modules in the EMCC program can bring security risks such as privacy leakage and information flow hijacking to EMCC; then, a generic framework of elastic mobile cloud computing that incorporates risk management is designed, which regards security risks as a cost of EMCC and ensures that the use of EMCC is. Finally, it is pointed out that the difficulty of risk management lies in risk quantification and sensitive module labeling. In this regard, risk quantification algorithms are designed, an automatic annotation tool for sensitive modules of Android programs is implemented, and the accuracy of the automatic annotation is demonstrated through experiments.

Corresponding Author:

Aws Hamed Hamad
Ministry of Higher Education and Scientific Research
Baghdad, Iraq
Email: aws.hamed@rdd.edu.iq

1. INTRODUCTION

Mobile networks are overgrowing. Wireless communication, social networking, gaming and entertainment, mobile finance/medical/education, reality enhancement and other emerging tasks are increasingly demanding on the performance of mobile devices [1]. However, mobile devices are limited by portability, ease of use, cost, and heat dissipation, and their computing, storage, and networking capabilities are inferior to those of their fixed counterparts; at the same time, the power constraint on the application time and scope of mobile devices is also increasingly prominent [2], and the power problem is even considered to be the main reason preventing smartphones from replacing personal communications services (PCs) [3]. The development of wireless network technologies such as code division multiple access 2000 (CDMA2000) provides the conditions for combining mobile devices and clouds. In mobile cloud computing (MCC), computing tasks can be performed outside the mobile device. The cloud provides various services for the mobile device, thus breaking the performance and power bottleneck of the mobile device and greatly expanding the range of mobile device usage [4]. The existing MCC is usually a C-S model [5]. The mobile tasks are divided into two parts: the client deployed in the mobile device gets the local information and the user input, and the server in the cloud pre-installs the corresponding software to provide the specified service. In this model of MCC, the division of tasks is fixed.

In order to utilize the computing resources in the cloud more flexibly and efficiently, a new MCC scheme is designed and implemented in papers [6]–[13]. The mobile device keeps all the data and codes needed to complete the task, and when it is not connected to the cloud, the mobile device completes the computation by itself; when it is connected to the cloud, the mobile device cuts the data and codes into multiple modules, migrates some of them to the cloud as needed, and employs the cloud to complete the computation. When connected to the cloud, the mobile device slices data and code into modules migrates some of them to the cloud as needed, and hires the cloud to do the computation to save power and increase computing speed. During the program's execution, the mobile device can dynamically adjust the task on demand and flexibly according to the specific execution environment (network conditions, power, task nature, and user mode). This new MCC model, mobile devices to the cloud computing resources are on-demand self-help, elastic variable so that such new mobile cloud computing for the elastic mobile cloud computing elastic mobile cloud computing (EMCC). Figure 1 gives a simple example: a program is divided into six modules, A, B, C, D, E, and F, where A and F are assigned to mobile devices, B, C, D, and E are assigned and migrated to the cloud, and then each module is executed in turn. In the execution process, the user requirements change, the allocation results change, and E is reallocated to the mobile device for execution. After the execution is completed, the mobile device displays the results.

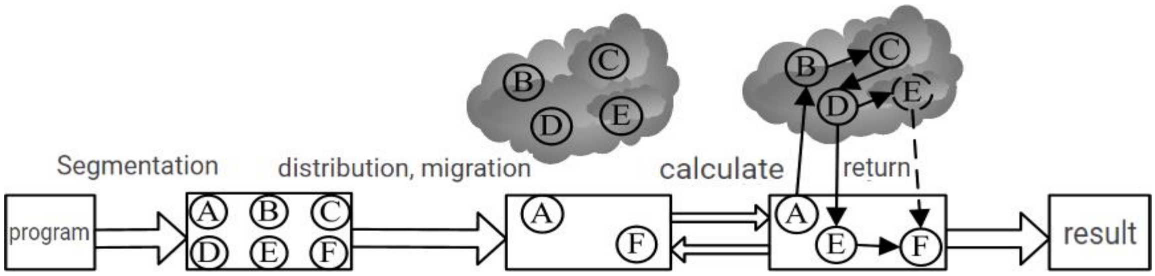

Figure 1. EMCC program execution process example

Security issues are seen as the primary obstacle to the development of cloud computing [14]. The mobility, openness, and instability of mobile devices make the security issues of mobile cloud computing (MCC) more complex than traditional cloud computing. The eavesdropping/interception of wireless channels, external and internal attacks faced by the cloud and theft/selling of information by cloud service providers (CSP) pose security threats to elastic mobile cloud computing (EMCC). For example, in Figure 1, modules B, C, D, and E are migrated to the cloud for execution. These modules may suffer from information theft and data modification threats during the migration and execution process. EMCC is still in its early stage of development, and existing research focuses on achieving fine-grained, user-transparent EMCC. In this paper, we analyze and summarize the EMCC security issues and point out that the leading way to improve EMCC security is to avoid the migration of "sensitive modules" with high confidentiality, integrity and availability requirements in high-risk scenarios possible. We design and implement a risk-controlled elastic mobile cloud computing module allocation framework based on this principle. This framework is generic and independent of the implementation details of each EMCC. The final analysis and experiments confirm the framework's usability described in this paper.

2. METHOD

There are many different EMCC schemes designed and implemented. However, most EMCC schemes generally use a common framework that includes the fundamental aspects of cloud environment construction, program partitioning, module assignment, module migration, program execution, and result return. At the same time, each EMCC faces similar security issues. A prerequisite for computing migration is constructing a cloud environment that is used to assist mobile devices with computing. The cloud environment here includes cloud computing resources and a cloud execution environment. Cloud computing resources can be built using public clouds, private clouds/dedicated clouds, local computers, and nearby mobile terminals. As listed in Table 1, various types of cloud computing resources have different characteristics.

Table 1. Types of cloud computing resources and their characteristics

Types of cloud computing resources	Corresponding plan	Advantage	Disadvantage
public cloud	[6]–[13]	Nearly unlimited resources	Network quality restricts service quality [15]; there is a problem of user trust
local cloud	[15]	Higher bandwidth, lower latency network	Requires high administrative costs; limited resources
local PC	[16]	Lower cost; Less latency; Convenience	Poor stability and security
local mobile terminal	[17], [18]	More convenient; Lower cost	It is easy to leak privacy; it is necessary to provide users with incentives to open computing resources

The cloud support for mobile devices can be provided by multiple cloud service providers or a hybrid of multiple resources [19], e.g., a resource pool can be jointly constructed by local PCs, local clouds, public clouds, and idle smartphones. to provide cloud computing and storage resources for mobile devices on-demand and flexibly. Cloud servers and mobile devices generally have different instruction set architectures (usually X86 for the former and ARM for the latter). They need to provide an environment in the cloud to support the execution of task modules, e.g., the Android x86 platform can be used to provide an execution environment for user EMCC programs. Resource sharing enables efficient use of resources: if resources are shared, the cloud only needs to provide the peak of the total demand of each node rather than the superposition of the peak demand of each node. Table 2. lists the various types of cloud execution environments and their characteristics.

Table 2. Types of cloud execution environments and their characteristics

Cloud execution environment	Example	Features
Shared module	Shared executable module [10]	High resource utilization, fast response speed, insecure
Shared execution environment	Modules for multiple users share an Android virtual machine [6], [7]	High resource utilization and poor security
Exclusive execution environment	One Android virtual machine per user [8], [15]	Low resource utilization, slow response speed, safer

The diversity of cloud computing resources and cloud execution environment brings various security issues. On the one hand, security flaws in the cloud may cause security problems. For example, virtual machine system vulnerabilities and multi-tenancy issues may lead to theft of user information and unavailability of services. Especially in the scenario of shared modules or shared execution environment, mobile application computing modules migrated to the cloud may need to share central processing unit (CPU), memory, clock, and the underlying Linux kernel and library files of the Android system with computing modules from other users, which provides convenience for attackers to implement side-channel attacks. On the other hand, malicious cloud resource owners can easily steal user data and disrupt the execution of user programs through the lower layer of information and resources. Program partitioning is the division of a task into executable modules and labeling the properties of each module separately. There are two main approaches to the cut-up in existing research.

- Redesigning the program architecture and program development methods suitable for running in MCC. For example, the EMCC application in [6] consists of one or more Weblets, a UI, and a declaration describing the application. A Web let is a program fragment that can run on a mobile device or in the cloud, perform a specific function independently, and provide an interface to interact with other Weblets or UI. Android application development scheme provides a compiler, and program developers mark @Remote for methods that can be migrated to the cloud to slice and dice Java programs at the granularity of methods.
- Analysis-based partitioning CloneCloud [7]. Analyzes existing programs to find the point where Android programs can be partitioned and then automatically partitions existing programs so that some threads can be migrated to the cloud for execution.

Compared with fixed devices, the execution environment of mobile applications is more variable. Network conditions, power, and CPU/memory utilization, vary from a mobile device to a mobile device. Therefore, the specific modules that need to be migrated to the cloud should be determined by the execution environment and can change with the execution environment. Assuming that the program is divided into *n* modules and there is a constraint that k modules cannot be migrated, there are 2^{n-k} module allocation methods. For example, in Figure 2, *n*=4, *k*=2, modules 1 and 4 cannot be migrated, so there are 4 module assignment methods. More allocation methods ensure the flexibility of task assignments. Module assignment is finding the best or better assignment among these 2^{n-k} possible assignments. The module assignment methods in some existing schemes are listed in Table 3.

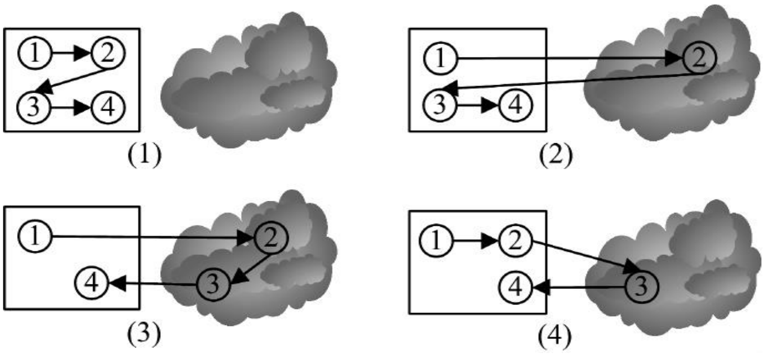

Figure 2. Module allocation method

Table 3. Example of a module allocation method in an existing EMCC scheme

Ref	Slicing granularity	Target	Considerations	Solution
[2]	Weblet	Electricity, capital, performance	Battery, network, load, performance	Machine learning, naive Bayesian models
[7]	Thread	Time, power	Network, CPU status	Consumption model, optimal selector
[8]	Method	Load, deadline, hardware environment	Network status, hardware status, and history	PowerTutor model, comparison
[11]	Custom module	Time	Module resource usage	Resource consumption graph
[10]	Components	CPU utilization, network status	maximum throughput	Genetic algorithm
[12]	Components	Mobile device performance, power; cloud consumption	Execution environment, cloud parallelism, and elasticity	Fuzzy logic models and empirical learning

Module migration is the migration of one or more modules that have been assigned to the cloud to the cloud, usually over a wireless network. Since the modules/parameters are often in plaintext during migration and are more susceptible to reference [2] or interference suppression [20] compared to twisted pair, fiber optic networks, wireless networks (phone services, short message service (SMS), 3G, Wi-Fi, and Bluetooth), attackers may use hardware to steal information, block information, and launch traditional denial of service attacks. The program execution is the sequential execution of each module assigned to the mobile device and the cloud. The flexible allocation of resources can be achieved by using virtual machines [11], increasing the computational speed by parallel computing [6], and increasing resource utilization by sharing resources [10]. After executing the modules in the cloud, the results are returned to the mobile device, which presents the final results to the user.

2.1. EMCC threat model

From the above analysis, it is clear that the existing EMCC schemes have different implementation details, but their basic implementation framework is more or less the same. Existing EMCCs face security threats such as system vulnerabilities and malicious codes in mobile devices, bandwidth/security issues in wireless channels, external/internal attacks on the cloud, and malicious cloud resource providers. Here, it is assumed that the module in the EMCC that is migrated to the cloud can perform all operations, such as sending SMS messages. Although the act of sending SMS may eventually need to be performed on the mobile device since the processing prior to sending SMS (e.g., recipient, SMS content generation) is done in the cloud, the action of "sending SMS" can be considered as done in the cloud environment from the security point of view. It will be subject to the wireless channel and cloud environment threats.

Mobile devices contain much valuable information for attackers, including user information, sensing information, system status, and user input. When sharing execution environments, other programs may steal information through underlying resources; malicious resource managers can steal information by logging memory, network, and CPU data. The vulnerability of wireless channels, hidden channels, attacks on virtual machines or virtual machine managers, insecure execution environments, and insecure resources make it difficult to protect the uploaded information.

In the cloud or wireless channel, an attacker may be able to modify the module, or information flow migrated to the cloud to compromise the integrity of the program execution process. By hijacking such program information flow/control flow, a cloud-based attacker can quickly achieve a variety of attacks. For example, suppose module A generates a set of parameters and module B uses them to modify the mobile terminal's system state or user information. If a is modified after being migrated to the cloud, it may cause B to modify the system or user information of the mobile terminal incorrectly. Such attacks may lead to virus worm propagation, system state/user information corruption, programs sending specific SMS messages to the wrong target or accessing the wrong network address, thus enabling denial attacks, phishing, chargeback attacks, and e-fraud. Authentication in existing security schemes, such as inter-module authentication in the literature [21], can only guarantee boot-time integrity, not run-time integrity, and cannot defend against such attacks.

Existing EMCC schemes mention some simple security measures. For example, the EMCC scheme in Salih *et al.* [6] specifies that Weblets involving local data can only be executed locally, and Clonecloud [7] cuts the thread back to the mobile device when local data is used. However, on the one hand, these approaches do not analyze which modules contain "local data", and the modules that may pose a threat to the security of the program are not only those that contain local data; on the other hand, not all modules that contain local data should be migrated to the cloud for execution, but rather an optimal solution should be found between user security and convenience. On the other hand, not all modules containing local data should not be migrated to the cloud for execution but should seek an optimal solution between user security and convenience [22].

2.2. A general framework for EMCC with controllable risk

To ensure that the use of EMCC is more beneficial than detrimental to users, security risks can be considered a cost in EMCC, and an EMCC framework is designed to integrate risk control to avoid the high-risk behavior of "migrating sensitive modules to the cloud" as much as possible. For mobile terminal users, installing programs with dangerous permission combinations, such as those with both *INTERNET* and *ACCESS_COARSE_I.O-CATION* permissions, is unsafe, but such programs are downloaded and installed in large numbers in Google Play [23]; rooting Android phones poses a security risk, but a large number of phones are still rooted. It can be seen that the demand for security is not absolute for ordinary users. Therefore, this paper assumes that users are willing to accept a specific security risk for functionality enhancement (users can give up using EMCC when dealing with very high-security requirement tasks) and consider security risk as an expense EMCC.

At different times, users may connect to many cloud resources through network connections with different bandwidths to complete computing in different cloud environments. For users, some of the task modules are migrated to the cloud to improve the execution speed of these modules and save the power of mobile devices; however, at the same time, module allocation, migration, and result return all require time and power and bring other expenses such as consuming network traffic and bearing security risks. Therefore, the benefits and expenses should be calculated. If the benefits are less than the expenses, EMCC should be abandoned, and the program execution should be done locally on the mobile device. In this paper, we design a generic, trying to ensure that the use of EMCC brings more benefits than its expenses; the overall framework of EMCC is shown in Figure 3.

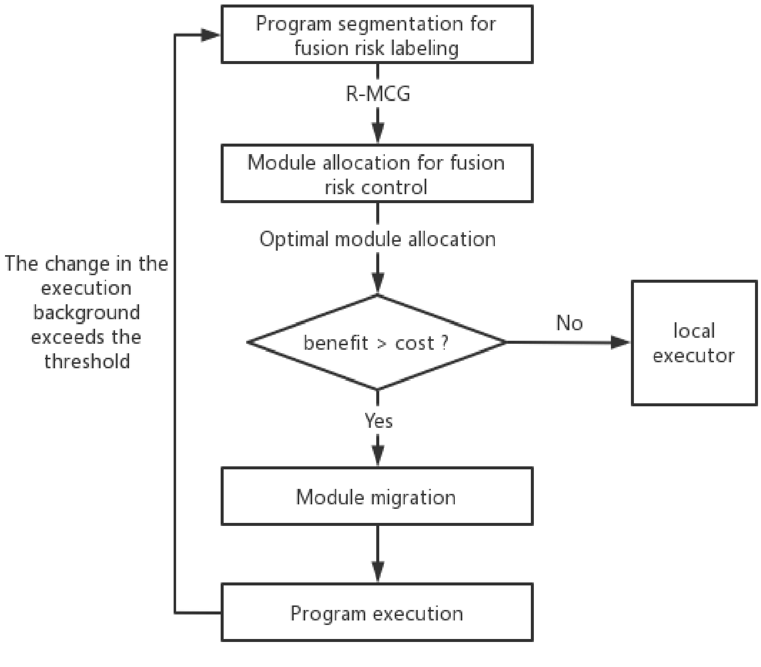
Program segmentation for fusion risk labeling
R-MCG
Module allocation for fusion risk control
Optimal module allocation
benefit > cost ?
No
local executor
Yes
Module migration
Program execution
The change in the execution background exceeds the threshold

Figure 3. EMCC common framework for integrated risk management

3. RESULTS AND DISCUSSION

The security risk associated with the migration of different modules varies. The specific risk value is related to the information contained in the module, the permissions involved, the program in which it resides, and the real-time execution environment. The risk is determined by the likelihood of occurrence of the hazard and the degree of the hazard. If these two aspects can be quantitatively assessed, the quantitative assessment of risk can be completed. Assuming that *R* represents the risk, *P* represents the probability of occurrence (threat probability), and *V* represents the hazard level, the above relationship can be expressed in (1).

$$R = V \times P \tag{1}$$

First, we analyze the degree of harm of privacy leakage, i.e., the degree of harm V_c of the risk of (confidentiality) R_c. For example, while user A thinks that the leakage of photo information is more harmful than the leakage of SMS content, user B may think the opposite; moreover, only users know the sensitivity of the information they input. Moreover, the degree of harm differs for different information leaks. For example, the harm of leaking SMS contact data is different. Therefore, in order to obtain the user's level of concern about the leakage of each information, it is necessary to understand the user's perception first, which can be obtained by referring to existing risk assessment methods [22], [23], such as statistical analysis, asking users, or machine learning.

The probability of threat occurrence is mainly associated with wireless channel type (WC), cloud execution environment, and cloud computing resource (CR). The probability of current wireless channel type, cloud execution environment, and cloud computing resource stealing information *IN* are P_{Cwc} *(IN)*, $P_{Ccp}(IN)$, and $P_{Cax}(IN)$, and the probability of modifying operation *OP* are $Pi_{wc}(OP)$, $Pi_{cp}(OP)$, and $Pi_{cx}(OP)$, respectively, then the information-stealing probability $P_C(IN)=P_{Cwc}(IN)+P_{Ccp}(IN)+P_{Ccx}(IN)$; information hijacking probability $P_i(OP)=P_{iwc}(OP)+P_{icp}(OP)+P_{icx}(OP)$. Its specific value can be derived by analyzing specific situations and historical statistical data [24].

Let the hazards caused by the theft of sensitive information *IN* and hijacking of sensitive operation *OP* in module *M* be $V_C(IN)$, $V_I(OP)$, respectively; assuming that no module containing *IN/OP* is migrated before M, the security risk caused by the migration of module M containing IN/OP is calculated as shown in (2). For the threat of information flow hijacking caused by module migration, i.e., integrity risk R_I the degree of harm V_I is calculated similarly as [25]–[29]. The key point to judge the degree of hazard is to analyze what kind of sensitive operation the obtained program performs.

$$R(M) = P_C(IN)V_C(IN) + P_I(OP)V_I(OP) \tag{2}$$

After defining the solution objectives, the module assignment problem becomes a multi-objective optimal solution. The weight of each solution objective can be decided according to the user's current power demand, time demand, and execution environment.

4. CONCLUSION

In recent years, various research institutions have designed and implemented various EM-CC solutions. This paper summarizes these solutions' common security problems and provides a familiar and easy-to-implement solution. This paper firstly establishes the common implementation framework of existing EMCC solutions (cloud environment construction, program partitioning, module assignment, module migration, program execution, result return). It then analyzes and summarizes the main security issues brought by EMCC from the perspective of mobile device users: privacy leakage and information flow hijacking, and points out that risk management can be used to control the security risks brought by EMCC program execution. The key to achieving risk management is to quantify the risks associated with each allocation scheme. To address this issue, we have designed a risk quantification method for EMCC module assignment and solved the main difficulty of this quantification method, i.e., the identification of sensitive modules. This framework can be used as a reference for the design of future EMCC schemes.

ACKNOWLEDGEMENTS

The authors are grateful to the Iraqi Ministry of Higher Education and Scientific Research (MOHESR) for technically supporting the current research. This work was supported by Al-Mustaqbal University College (MUC-E-0122).

REFERENCES

[1] A. S. Al Ahmad, H. Kahtan, Y. I. Alzoubi, O. Ali, and A. Jaradat, “Mobile cloud computing models security issues: a systematic review,” *Journal of Network and Computer Applications*, vol. 190, Sep. 2021, doi: 10.1016/j.jnca.2021.103152.

[2] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, “To offload or not to offload? the bandwidth and energy costs of mobile cloud computing,” in *Proceedings IEEE INFOCOM*, Apr. 2013, pp. 1285–1293, doi: 10.1109/INFCOM.2013.6566921.

[3] A. H. Shatti, H. A. Hasson, and L. A. Abdul-Rahaim, “Automation conditions of mobile base station shelter via cloud and IoT computing applications,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 5, pp. 4550–4557, Oct. 2021, doi: 10.11591/ijece.v11i5.pp4550-4557.

[4] L. Pallavi, A. Jagan, and B. T. Rao, “ERMO2 algorithm: an energy efficient mobility management in mobile cloud computing system for 5G heterogeneous networks,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 3, pp. 1957–1967, Jun. 2019, doi: 10.11591/ijece.v9i3.pp1957-1967.

[5] Z. A. Jaaz, I. Y. Khudhair, H. S. Mehdy, and I. Al Barazanchi, “Imparting full-duplex wireless cellular communication in 5G network using apache spark engine,” in *8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Oct. 2021, pp. 123–129, doi: 10.23919/EECSI53397.2021.9624283.

[6] S. Q. Salih *et al.*, “Integrative stochastic model standardization with genetic algorithm for rainfall pattern forecasting in tropical and semi-arid environments,” *Hydrological Sciences Journal*, vol. 65, no. 7, pp. 1145–1157, May 2020, doi: 10.1080/02626667.2020.1734813.

[7] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “Clonecloud: elastic execution between mobile device and cloud,” in *Proceedings of the sixth conference on Computer systems*, 2011, pp. 301–314, doi: 10.1145/1966445.1966473.

[8] S. Kosta, A. Aucinas, Pan Hui, R. Mortier, and Xinwen Zhang, “ThinkAir: dynamic resource allocation and parallel execution in the cloud for mobile code offloading,” in *Proceedings IEEE INFOCOM*, Mar. 2012, pp. 945–953, doi: 10.1109/INFCOM.2012.6195845.

[9] S.-E. Chafi, Y. Balboul, S. Mazer, M. Fattah, and M. El Bekkali, “Resource placement strategy optimization for smart grid application using 5G wireless networks,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 4, pp. 3932–3942, Aug. 2022, doi: 10.11591/ijece.v12i4.pp3932-3942.

[10] N. J. Qasim, S. M. Mohammed, A. S. Sosa, and I. Albarazanchi, “Reactive protocols for unified user profiling for anomaly detection in mobile Ad Hoc networks,” *Periodicals of Engineering and Natural Sciences (PEN)*, vol. 7, no. 2, pp. 843–852, Jul. 2019, doi: 10.21533/pen.v7i2.497.

[11] Y. K. Salih, O. H. See, S. Yussof, A. Iqbal, and S. Q. Mohammad Salih, “A proactive fuzzy-guided link labeling algorithm based on MIH framework in heterogeneous wireless networks,” *Wireless Personal Communications*, vol. 75, no. 4, pp. 2495–2511, Apr. 2014, doi: 10.1007/s11277-013-1479-z.

[12] J. Shen, D. Liu, J. Shen, Q. Liu, and X. Sun, “A secure cloud-assisted urban data sharing framework for ubiquitous-cities,” *Pervasive and Mobile Computing*, vol. 41, pp. 219–230, Oct. 2017, doi: 10.1016/j.pmcj.2017.03.013.

[13] S. Fugkeaw, “A secure and efficient data sharing scheme with outsourced signcryption and decryption in mobile cloud computing,” in *2021 IEEE International Conference on Joint Cloud Computing (JCC)*, 2021, pp. 72–79, doi: 10.1109/JCC53141.2021.00024.

[14] I. Al-Barazanchi *et al.*, “Remote monitoring of COVID-19 patients using multisensor body area network innovative system,” *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–14, Sep. 2022, doi: 10.1155/2022/9879259.

[15] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for VM-based cloudlets in mobile computing,” *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct. 2009, doi: 10.1109/MPRV.2009.82.

[16] E. Y. Chen and M. Itoh, “Virtual smartphone over IP,” in *IEEE International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM)*, Jun. 2010, pp. 1–6, doi: 10.1109/WOWMOM.2010.5534992.

[17] X. Lu, Z. Pan, and H. Xian, “An efficient and secure data sharing scheme for mobile devices in cloud computing,” *Journal of Cloud Computing*, vol. 9, no. 1, Dec. 2020, doi: 10.1186/s13677-020-00207-5.

[18] S. Zouaidi, A. Belghith, and I. Lengliz, “SVCF: Secure vehicular cloud framework,” in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, Jun. 2020, pp. 412–419, doi: 10.1109/IWCMC48107.2020.9148574.

[19] A. Al-Omary, “A secure framework for mobile cloud computing,” in *2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Sep. 2019, pp. 1–6, doi: 10.1109/3ICT.2019.8910294.

[20] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, “Denial of service attacks in wireless networks: the case of jammers,” *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, pp. 245–257, 2011, doi: 10.1109/SURV.2011.041110.00022.

[21] X. Zhang, J. Schiffman, S. Gibbs, A. Kunjithapatham, and S. Jeong, “Securing elastic applications on mobile devices for cloud computing,” in *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, 2009, doi: 10.1145/1655008.1655026.

[22] H. Li, Q. Huang, and W. Susilo, “A secure cloud data sharing protocol for enterprise supporting hierarchical keyword search,” *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1532–1543, May 2022, doi: 10.1109/TDSC.2020.3027611.

[23] A. Mylonas, M. Theoharidou, and D. Gritzalis, “Assessing privacy risks in android: a user-centric approach,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8418, Springer International Publishing, 2014, pp. 21–37.

[24] M. B. Monir Mansour, T. Abdelkader, M. H. AbdelAziz, and E.-S. M. EI-Horbaty, “A trust evaluation scheme of service providers in mobile edge computing,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, pp. 2121–2138, Apr. 2022, doi: 10.11591/ijece.v12i2.pp2121-2138.

[25] J. Al-Muhtadi, K. Saleem, S. Al-Rabiaah, M. Imran, A. Gawanmeh, and J. J. P. C. Rodrigues, “A lightweight cyber security framework with context-awareness for pervasive computing environments,” *Sustainable Cities and Society*, vol. 66, Mar. 2021, doi: 10.1016/j.scs.2020.102610.

[26] H. R. Abdulshaheed, H. H. Abbas, E. Q. Ahmed, and I. Al-Barazanchi, “Big data analytics for large scale wireless body area networks; challenges, and applications,” in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 127, Springer International Publishing, 2022, pp. 423–434.

[27] S. S. Oleiwi, G. N. Mohammed, and I. Al_Barazanchi, “Mitigation of packet loss with end-to-end delay in wireless body area network applications,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, pp. 460–470, Feb. 2022, doi: 10.11591/ijece.v12i1.pp460-470.

[28] N. S. Mohd Pakhrudin, M. Kassim, and A. Idris, “A review on orchestration distributed systems for IoT smart services in fog computing,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 2, pp. 1812–1822, Apr. 2021, doi: 10.11591/ijece.v11i2.pp1812-1822.

[29] T. Francis, “A comparison of cloud execution mechanisms fog, edge, and clone cloud computing,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 6, pp. 4646–4653, Dec. 2018, doi: 10.11591/ijece.v8i6.pp4646-4653.

BIOGRAPHIES OF AUTHORS

Aws Hamed Hamad received a B.Sc. in computer science from Mustansiriyah university. Then, he got a M.Sc. in Artificial intelligence from Wuhan University, China in 2019. He has been working for the Iraqi Ministry of Higher Education and Scientific Research at Research and Development department. Recently, he has given lectures at university regarding Artificial intelligence. He has some publications in international journals. His work interests include artificial intelligence, machine learning, deep learning, and optimization. He can be contacted at aws.hamed@rdd.edu.iq.

Adnan Yousif Dawod obtained a Bachelor's degree in Computer Software Engineering from Kirkuk Technical College, Kirkuk, Iraq in 2003, and a Master's degree in Computer Engineering from Sam Higginbottom Institute, Allahabad, India in 2014. During 2004-2012 he worked as an engineer and administrator of the Computer and Internet Division in College of Nursing, University of Kirkuk. During the period 2012-2020, he worked as a lecturer and in charge of the Computer and Internet Division at the College of Nursing, University of Kirkuk. During the period 2020 until now, he worked as a lecturer and head of the Continuing Education Division at the College of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq. He can be contacted at email: adnanalshef@uokirkuk.edu.iq.

Mohammed Fakhrulddin Abdulqader received his Bachelor's degree in Computer Software Engineering from Kirkuk Technical College/Kirkuk Iraq in 2003, and his Master's degree in Computer Engineering from Sam Higginbottom Institute, Allahabad-India, 2014. During the period 2004-2006 he worked as an engineer in Presidency of the University of Kirkuk and then in 2006 he joined the College of Engineering, University of Kirkuk, Kirkuk, Iraq to work as an engineer and in charge of the Computer and Internet Division at the College of Engineering, University of Kirkuk. During the period 2006-2020, he worked as a lecturer and head of the Computer and Internet Division at the College of Engineering, University of Kirkuk, Kirkuk, Iraq. During the period 2020-2022, he worked as a lecturer and official of the Computer and Internet Division at the College of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq. During the period 2022 until now, he works as a lecturer, official and director of scientific affairs at the College of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq. He can be contacted at email: mohammed_mf81@uokirkuk.edu.iq.

Israa Al_Barazanchi received her Bachelor of Computer Science (BCS) from Department of Computer Science, Baghdad college of economic science university-Iraq-Baghdad in June 2002. In January 2010, she entered the master's program at the Faculty of Information and Communication Technology, Graduate School of Computer Science (Internetworking Technology), Universiti Teknikal Malaysia. She is Currently a student Doctor of Philosophy in Information and Communication Technology. She is a lecturer in Computer Engineering Techniques Department. Editor-in-chief for international union of universities journal. Member of editor board in many scopus journals for computer science field. Head of researcher group. Member in many conferences panel and communications events. She is a reviewer of various high impact factor journals. Her research activities are: WBAN, WSN, WiMAX, WiFi on vehicular ad-hoc networks (VANETs), communications, networking, signal processing, IoT, IoMT, smart healthcare systems, blockchain. She can be contacted at israa.albarazanchi@baghdadcollege.edu.iq and israa44444@gmail.com.

Hassan Muwafaq Gheni received his Bachelor (B.Sc) of Electrical & Electronic Engineering from Department of Electrical Engineering, Babylon University-Iraq-Hilla in June 2016. In February 2018, he entered the master's program at the Faculty of Electrical and Electronic Engineer, Universiti Tun Hussein Malaysia. He is a lecturer at Al-Mustaqbal university college/ Department of Computer Techniques Engineering His research interest is optical communication, IoT, wireless sensor network, communications, V2V system, artificial intelligent. He can be contacted at hasan.muwafaq@mustaqbal-college.edu.iq.