Scientific Research Publishing

Communications and Network, 2020, 12, 155-173
https://www.scirp.org/journal/cn
ISSN Online: 1947-3826
ISSN Print: 1949-2421

A Hybrid IoT Security Model of MQTT and UMA

Khalid S. Aloufi[1], Omar H. Alhazmi[2]

[1]Department of Computer Engineering, Taibah University, Medina, Saudi Arabia
[2]Department of Computer Science, Taibah University, Medina, Saudi Arabia
Email: koufi@taibahu.edu.sa, ohhazmi@taibahu.edu.sa

How to cite this paper: Aloufi, K.S. and Alhazmi, O.H. (2020) A Hybrid IoT Security Model of MQTT and UMA. *Communications and Network*, **12**, 155-173.
https://doi.org/10.4236/cn.2020.124008

Received: October 9, 2020
Accepted: November 16, 2020
Published: November 19, 2020

Copyright © 2020 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/

Open Access

Abstract

IoT applications are promising for future daily activities; therefore, the number of IoT connected devices is expected to reach billions in the coming few years. However, IoT has different application frameworks. Furthermore, IoT applications require higher security standards. In this work, an IoT application framework is presented with a security embedded structure using the integration between message queue telemetry transport (MQTT) and user-managed access (UMA). The performance analysis of the model is presented. Comparing the model with existing models and different design structures shows that the model presented in this work is promising for a functioning IoT design model with security. The security in the model is a built-in feature in its structure. The model is built on recommended frameworks; therefore, it is ready for integration with other web standards for data sharing, which will help in making IoT applications integrated from different developing parties.

Keywords

IoT, MQTT (Message Queuing Telemetry Transport),
UMA (User-Managed Access), Network Security, Smart City

1. Introduction

Over the past few years, one of the remarkable emerging technologies is the Internet of Things (IoT). It has vast varieties of possible applications, with a wide range of requirements in terms of security and performance. Several networking protocols are designed with performance in mind, then security is added later. Thus, security is often considered a burden and an overhead to the original protocol. The original non-secure protocol requires less communication and encryption overhead. Moreover, an IoT environment mainly consists of constrained devices, with limited resources and capabilities, which makes perfor-

mance vital for the environment.

There is an essential need for a protocol that distributes the load and decentralizes security in order to have a scaleable IoT environment and satisfy the performance and security requirements at the same time.

One of the widely accepted IoT protocols is Message Queuing Telemetry Transport (MQTT). MQTT is an ISO standard (ISO/IEC PRF 20922) [1]. It is a lightweight, publish-subscribe network protocol that transports messages between devices. MQTT typically runs over TCP/IP. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited.

MQTT does not provide security by itself. Therefore, there have been some improvements to MQTT to improve its security; however, the overhead remains a critical issue in an environment with constrained devices. On the other hand, User-Managed Access (UMA) provides security.

User-Managed Access (UMA) is an OAuth-based access management protocol standard. Version 1.0 of the standard was approved by the Kantara Initiative on March 23, 2015 [2].

OAuth is an open standard for access delegation, commonly used as a way for Internet users and applications to grant websites or applications access to their information on other websites. It leaves the power with the owner to control authorization. It provides privacy and consent implications for web applications and the Internet of Things (IoT).

The purpose of this work is to enhance IoT security by integrating non-centralized authentication and at the same time maintaining an acceptable level of performance given that most Internet devices are constrained and has limited computation capabilities.

In the rest of this paper, we present some related work (Section 2), then the background of MQTT and UMA in Section 3. Then in Section 4, we propose a hybrid model by mapping MQTT and UMA. In Section 5, by simulation, we evaluate the proposed secure MQTT/UMA hybrid system. Then the simulation experiment results are presented, discussed and analyzed in Section 6; finally, a conclusion and final remarks are summarized, and some future research directions are presented.

2. Related Work

Internet of Things deployment remains a major challenge; hence, many have suggested using the cloud as a platform for the Internet of Things [3]. It has also been suggested that IoT sensors can be integrated into the cloud [4]; moreover, the business model can be built around providing Sensing as a Service module (SenaaS) [5]. Neven Nikolov and Ognyan Nakov have suggested using MQTTS and SSL/SSH encryption for IoT device [6]. Such a scheme can be effective; however, the performance impact on IoT should be further investigated. Others have also proposed some enhancements to MQTT by using Attribute-Based En-

cryption (ABE) to secure some aspects of MQTT [7] [8].

Perira *et al.* have listed challenges of IoT, which are automated sensor configuration and context discovery context acquisition, modeling, reasoning, and distribution selection of sensors in sensing-as-a-service (SenaaS) models security, and privacy-trust and context sharing [9]. Dai, Zheng and Zhang have proposed Blockchain of Things (BCoT) to secure IoT devices. Their model seems very secure. However, the high demand for resources can make it infeasible with the current constrained IoT device [10]

Niruntasukrat *et al.* have modified the OAuth 1.0a framework to accommodate the restrictions of IoT devices [11]. Ullah, Ahmed and Kim have proposed information-centric networking [12]. They have elaborated on the importance of the fog layer, as a practical and gap filler layer in any deployment of the IoT/Cloud environment.

Alhazmi and Aloufi have shown how deploying IoT devices using fog computing has an evident performance advantage over using cloud [13]. Moreover, in [14] they have suggested an IoT implementation by mapping MQTT protocols over the cloud-fog by placing the broker on the fog; over REST architecture style, the results improved performance and reliability. On Access control in IoT Cruz-Piris, Rivera, Mersa-Maestre, De la Hoz, and Velasco have proposed an access control mechanism for IoT using Oauth 2.0 with MQTT [15].

3. Background

3.1. MQTT Protocol

There are different IoT protocols, such as MQTT and CoAP. While CoAP is highly competitive with MQTT. CoAP has a mechanism of exploration and observation; however, MQTT is much simpler to implement and much more popular [16] [17] [18]. As shown in **Figure 1**, MQTT consists of clients and a broker. The central part is the broker, the broker manages messages and transactions between clients. Clients are either a subscriber or a publisher. The payload of messages transmitted between clients contains data, mainly a subject and its value. The publisher sends messages to the broker when it has an update message or a periodic message. The broker sends the messages to the subscribers of a specific subject.

Figure 2 shows an MQTT transaction flow. **Table 1** shows the IoT stack model for MQTT. With correspondence of the TCP/IP model, MQTT is one of the main IoT protocols because its messages are transmitted over TCP connection, in contrast to some other protocols, such as CoAP, which are transmitted over UDP. MQTT takes full advantage of the TCP/IP model, making application development and understanding of the protocol trivial. Also, MQTT integration over TCP help in the integration of MQTT with other protocols that share the same stack.

3.2. UMA Protocol

OAuth2 is an open protocol to allow secure authorization of application without

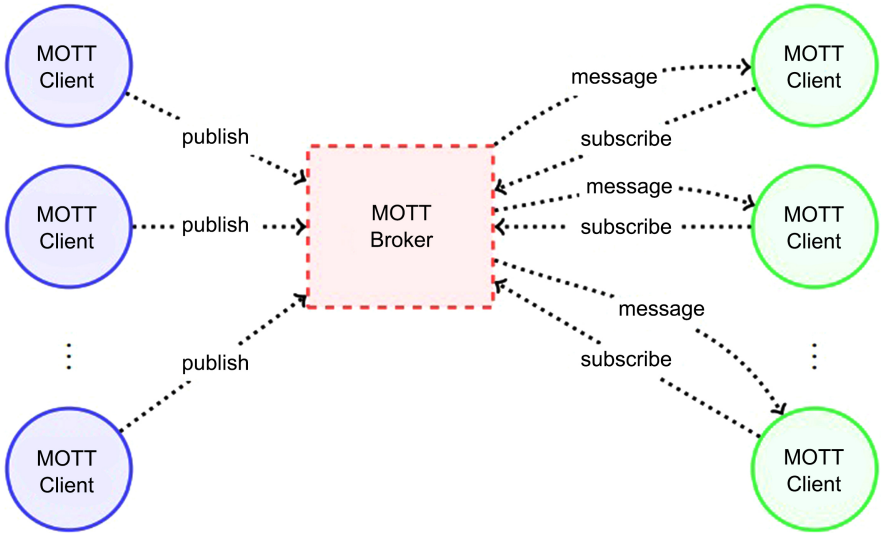

Figure 1. MQTT protocol model.

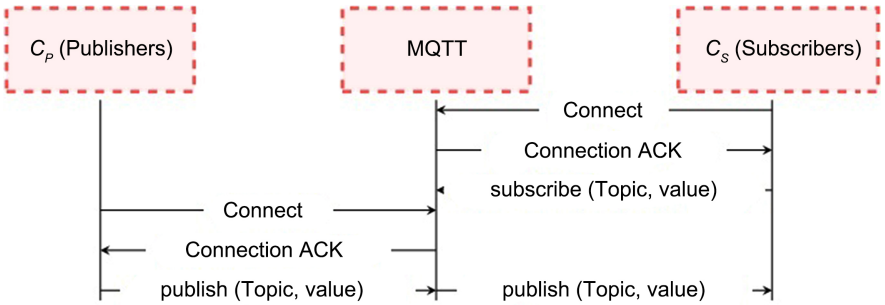

Figure 2. MQTT publish and subscribe transaction flow.

Table 1. Internet of things stack.

Layer	Table Column Head		
	OSI Layers	TCP/IP	IoT
7	Application	Application	MQTT
6	Presentation		
5	Session		
4	Transport	Transport	TCP
3	Network	Internet	IP
2	Data Link	Network Access	IEEE 802.15.4 MAC
1	Physical		IEEE 802.15.4 PYS

providing the password [19]. User-Managed Access (UMA) is a profile of OAuth 2.0. UMA defines how different entities communicates together. UMA consists of resource owner (RO), resource server (RS), authorization server (AS), client and the requesting party (RP). The components work together to provide secure access to resources using protection API, authorization API and tokens to access

protected resource.

The resource owner controls resource access by clients. The client is operated by a requesting party. Resources are hosted by a resource server. An authorization server has roles defined by the resource owner to access resources. For example, a resource owner can grant access to an application for one-time access to a resource following some roles defined by the owner and managed by the authorization server.

There are three phases of the UMA of resources as shown in **Figure 3**, which are protection, authorization and access [19]. **Figure 3** shows the transaction of the three phases. In phase one, the resource owner is managing resources at the resource server ("A"). The resource owner controls the authorization server ("C"), which provides the resources with protection API (B).

The protection API require a protection API token (PAT) to access a resource.

In phase two, the authorization phase, the client gets access to a set of resources in the resource server after being authorized by the authorization server. The authorization API is using an authorization API token (AAT) to get a requesting party token (RPT) ("D").

Finally, in phase three, the client accesses a resource in the resource server using an RPT ("E").

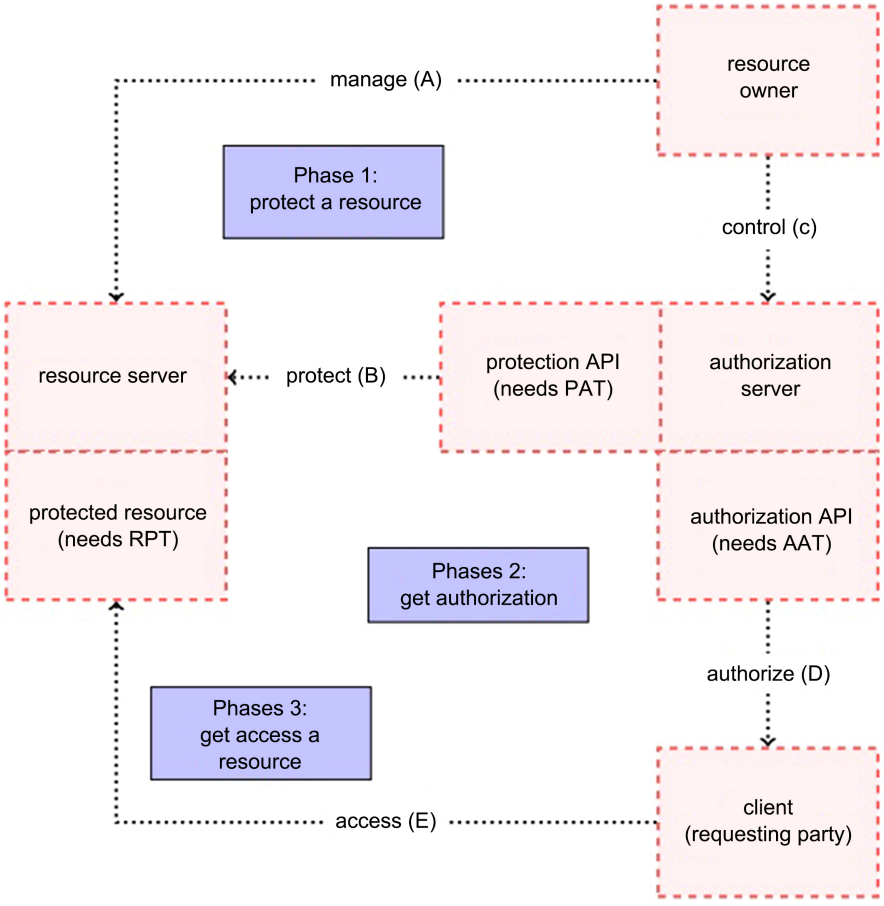

Figure 3. The three phases of the UMA profile Oauth [19].

Figure 4 shows the summarized transaction of the model. The client connects with the RS.

It gets the resource, then the client connects to the AS to get grant access and finally the client can get the resource from the RS. This transaction is updated from the original one since the UMA model is updated and the client has no direct connection with the RO [20].

The UMA model has different APIs and tokens to be used. Table 2 shows the list of APIs used, tokens, the issuer and the user of the tokens. Using the protection API, a Protection API Token (PAT) is issued by the AS to the RS to access the protection API. The RS is then issued using the RPT to protect a specific resource for a specific owner.

An Authorization API Token (AAT) is issued by the AS using the Authorization API when the RP contacts the AS with a well-formatted request as recommended by UMA [19].

The RP will use the token to contact the Authorization API again to get a Requesting Party Token (RPT), which is the token that the RP is working to grant based on ATT.

Figure 5 shows the detailed UMA transaction [21] [22]. The RO logs in to the RS to share a R. Then, the RS connects with the AS to get a PAT for the R. The AS replies with a PAT with the R, RS and RO information. After that, when the client (RP) requests access to the R at the RS, the client requires the access credentials. For this reason, the client is communicating with the AS to get the RPT and AAT for the specific R. Before the access permission, the AS set the access permissions rules. Finally, the client gets the R with a valid RPT. The UMA data should be exchanged in JSON format [21].

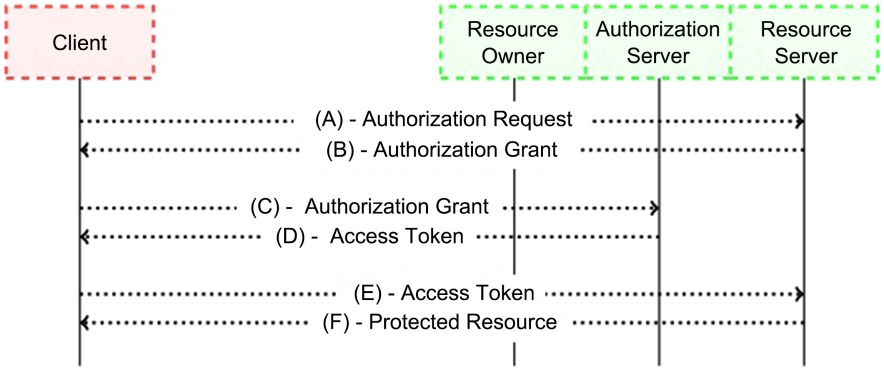

Figure 4. Summary of UMA transactions.

Table 2. UMA API and tokens.

Token	API	Issued By	Issued To	To Access
Authorization Party Token (AAT)	A API	AS	RP (Client)	A API
Protection API Token (PAT)	P API	AS	RS	P API
Requesting Party Token (RPT)	A API	AS	RP (Client)	R in RS

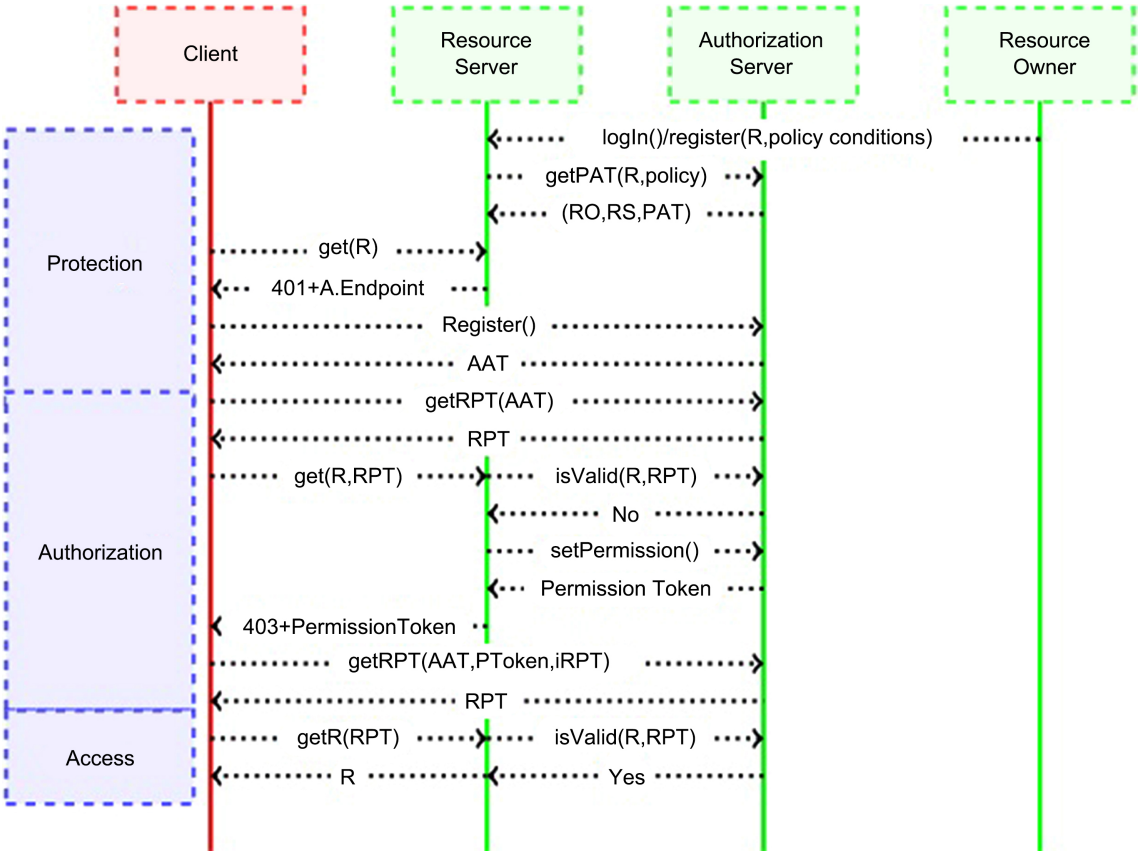

Figure 5. Detailed UMA transactions.

4. MQTT/UMA Hybrid Model

The proposed model is a system composed of two known systems, which are the UMA and MQTT. In the following sections, the model is developed. The first step is to do a mapping between the functions of the MQTT and UMA protocols. Then, in the next step, the model is proposed. The proposed model considers both the functionality of MQTT and UMA to provide the features of both protocols. Therefore, the model extends the area of application of UMA to reach IoT devices that works with MQTT protocols.

The model proposed should increase the security level of small to large scale IoT application in smart city or smart building applications.

4.1. UMA and MQTT Mapping

The model requires mapping between the functionality of both UMA and MQTT to work as one system.

This section shows the mapping of the function of both protocols, UMA and MQTT.

Table 3 shows the mapping required for the model between MQTT and UMA to work effectively. UMA consists of authorization server (AS), resource server (RS), resource owner (RO), client and requesting party (RP). MQTT consists of MQTT broker, MQTT publisher and MQTT subscribers.

Table 3. Mapping UMA and MQTT.

MQTT	UMA
P1	None
MB1/P2	RO
S1/MB2	RS
none	Client
S2	RP
none	Client
Topic/subject	R

The AS, RS, RO, client and RP are mapped with the MQTT broker, MQTT publisher and MQTT subscribers.

The model, which will be detailed in Section 4.2, consists of one UMA and two MQTT. MQTT manages topics which is considered an R in UMA. The first MQTT consists of MQTT Broker 1 (MB1), subscriber (S1) and publisher (P1). The second MQTT consists of MQTT Broker 2 (MB1), subscriber (S2) and publisher (P2). The model works by having P1 publish topics from IoT devices. MB1 gets the published topics from P1. After-that, MB1 publishes topics to MB2. S1 is MB2 and P2 is MB1. After the topic is published from P1 to MB1, MB2 gets the published topics from MB1. Finally, MB2 publishes topics to S2, the subscriber in the second MQTT.

To add UMA to the model, MB1 is mapped with the RO because of the similar functionality of both since both have resources to share. The S2 is mapped with a RP since S2 subscribes to a topic with the MB2. The S2 is considered as a RP because it will require use of a client to access a resource published by a MB2. The MB2 works as a RS because of the similar functionality of holding the resources access permissions. For the model to be practical and logical for implementations, the AS and client are considered two different standalone fog servers.

One of the main transactions added by the MQTT broker is the publishing of any new topic value. Now the RS has the resource and has its value. The resource is available for any successful RP access. However, when a new value of a resource is received, RS must update the resource and publish the resource to its subscribers. The integration between the functionality of the MB2 and RS is the main contribution of the model.

4.2. The MQTT/UMA Hybrid Model

Figure 6 shows the applicable network model proposed in this work. The model consists of the P1 as a publisher, S2/RP as a subscriber, RS/MB2/S1, MB1/P2/RO, client and AS. As will be shown later, most of the transaction messages of the proposed model are between the three main entities, which are S1/MB2/RS, client and AS. Therefore, to increase the performance of the model, RS/MB2/S1, Client and AS are in the fog layer.

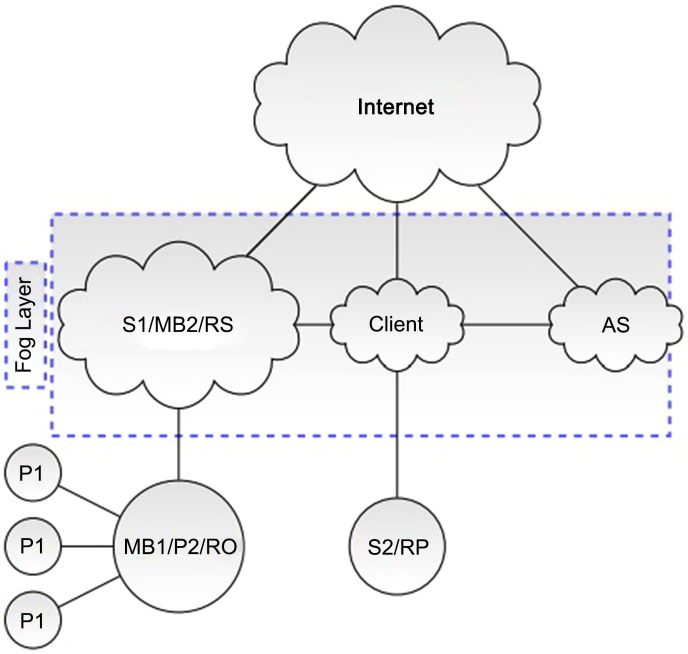

Figure 6. The hybrid model architecture.

As known and shown earlier in Section 4.1, the basic MQTT model consists of an MQTT broker, subscribers and publishers. In the model two MQTT models are joined to make a new MQTT model. Technically there are no P2 and S1 in the system. P2 and S1 functionality are integrated in MB1 and MB2, consequently.

S2/RP gets the messages from the fog layer, which is expected to provide more connectivity in availability and connection speed compared to the first mode as well as more security enhancement with UMA model.

MB2 is in the fog layer, which is connected directly with MB1. The model can be extended by having the MB2 connected to more than one MB1. Furthermore, each MB1 could be using one or more methods for connectivity to the fog layer.

The fog layer provides more performance and security of the high Quality of Service (QoS) required of a large-scale application [23]. One of the expected benefits of using the fog cloud is to decrease the response time between the subscribers and publishers.

The publishers notify the broker for any updates and the broker notifies the subscribers. The broker also has the log of each publisher in case it is needed by subscribers or for more statistical computations, such as the average value of a specific publisher or timing values, such as the last time a specific publisher updated its value.

5. Model Evaluation

In this section, we describe the evaluation methodology, which consists of four parts: the simulation setup, the simulation presentation; the results and discussion are presented afterward.

5.1. System Initial Configuration

This section shows the results and discussion of the simulation experiments of the IoT model shown earlier in Section 4.2.

As shown in Figure 7, the model contains P1, MB1/P2/RO, S1/MB2/RS, client, S2/RP and AS.

While P1 is in direct connection to MB1/P2/RO, there is no direct connection between MB1/P2/RO and S2/RP. S2/RP gets the published topic from S1/MB2/RS, which is well established in security and performance in the fog layer. MB1/P2/RO is expected to be a private property node in a home or a building at a security level that should not establish connection with general connections, such as RPs. Also, S1/MB2/RS is assumed to be connected with MB1/P2/RO with normal Internet connection.

This is an increase in security without added security overhead for MB1/P2/RO messages. Effectively, the gained security overhead for messages is added to messages between S1/MB2/RS and MB1/P2/RO and between S1/MB2/RS and S2/RP.

In Section 3.2, Figure 5 shows the detailed UMA transactions to add security for general systems. Figure 7 shows messages exchange between UMA and MQTT units to provide a reliable IoT system that uses UMA as an added security layer. Most messages transactions in the model are between the RS, client and AS. Therefore, S1/MB2/RS, AS and client are allocated in the model in the fog layer with high speed connection with each other to increase QoS.

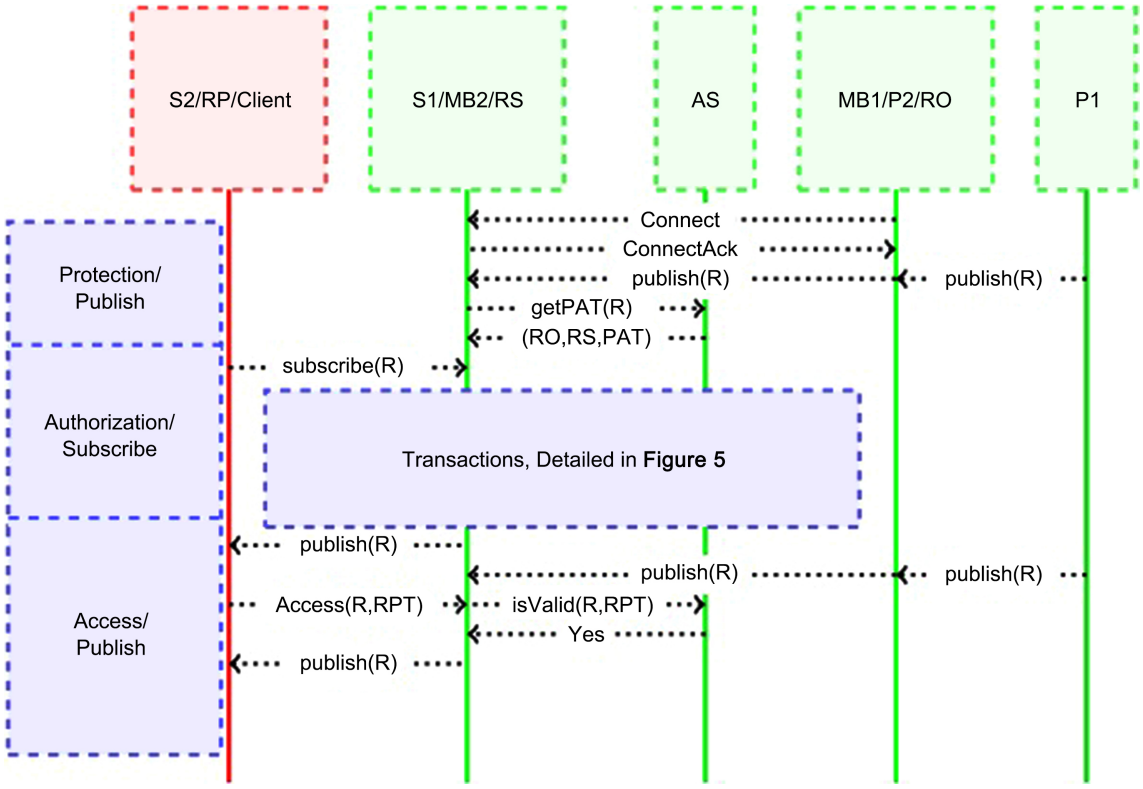

Figure 7. MQTT/UMA subscribe transactions.

5.2. Evaluating the Model

The model evaluation is done through simulation of the MQTT/UMA hybrid system. To configure the simulation model, the system is configured as follows. The wireless connection with any entity in the fog layer is 120 Mbps, assumed by a ping to the MQTT broker at test.mosquitto.org, and also tested by Jaloudi [24]. MB1/P2/RO can get from 10 - 100 Mbps over 802.11 g protocol or wired connection. Therefore, MB1/P2/RO can connect with several IoT devices at once, with a high data rate from IoT. However, the sending rate of each IoT device is much lower, allowing the broker to receive data from several IoT devices. Each IoT device is equipped with Zigbee, using an IEEE 802.15.4 antenna, with a rate of 250 kbps and the maximum message size is 127 bytes, according to the Zigbee Specification [25]. As a result, the transmission time of one message is about 4 ms.

Fog servers are assumed to be equipped with powerful nodes, such as Intel Xeon W-3275M @ 2.50GHz. In the fog server, the mean time required to access a database at the S1/MB1/RS is 13 ms. The mean time required to query the database is about 10 ms. The mean processing time at the S1/MB2/RS, AS, and client is 10 ms. IoT devices represented by $P1$ are publishing data with the Poisson arrival process.

A ping between two servers in Europe, assumed to be servers in the fog layer, requires between 3 ms and 100 ms, depending on the location of the fog servers. In this work, it is assumed that the fog server is in the same area, such as a country. Therefore, the assumed time is about 3 ms to exchange a message between two fog servers. The tested ping between two fog servers in different areas, like the US and Europe, is about 200 ms. From the simulation experiments, the arrival rate of data from each of the IoT devices is 127 bytes per second, which is one reading of a subject data keeping the data size minimal to avoid fragmentation.

Table 4 shows the transaction time between the different nodes of the model, and the processing time at each node. **Table 4** is the conclusion of the configuration of the model. The table is used to configure the simulation. Processing time at the IoT device is defined by T_{P1}. The table symbols are shown in **Figure 8**.

The transmission time between IoT device *P*1 and MQTT Broker *MB*1/*P*2/*RO* is defined by $T_{P1xMB1/P2/RO}$. The same applies to other transactions. The symbols shown in **Table 3** and **Table 4** are as follows: *P*1 is the IoT devices. *MB*1 is the first broker connected directly with *P*1 and connected with the *S*1 node. as the subscriber. *MB*1 shares the node with *P*2 and *RO*. The model is mutually inclusive between the two-broker system of *MB*1 and *MB*2. *P*2 is the publisher of the second broker system. As mentioned, *P*2 share the node with *MB*1 and *RO*.

*MB*2 is the second broker and share the node with *S*1 and *RS*. *MB*2 connects to *S*2 through the Client. *S*2 shares its node with *RP*.

$$1\times T_{MP1/P2/RO}+13\times T_{AS}+10\times T_{S1}+10\times T_{Client}+1\times T_{MB1/P2/RO\times S1/MB2/RS}+6\times T_{S1/MB2/RS\times AS}+4\times T_{Client\times S1/MB2/RS}+6\times T_{Client\,AS}=1207\,\text{ms} \quad (1)$$

$$2\times T_{AS}+2\times T_{S1/MB2/RS}+2\times T_{Client}+2\times T_{S1/MB2/RS\times AS}+2\times T_{Client\times S1/MB2/RS}=164\,\text{ms} \quad (2)$$

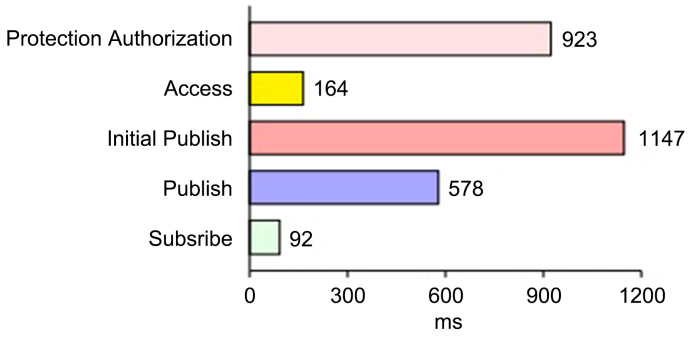

Figure 8. Processing time of each phase.

Table 4. Time required for processing and transaction.

Parameter	t (ms.)
T_{P1}	10
$T_{MB1/P2/RO}$	100
T_{AS}	13 + 10 + 10
$T_{S1/MB2/RS}$	13 + 10 + 10
$T_{S2/RP}$	100
T_{Clinet}	10
$T_{P1xMB1/P2/RO}$	4
$T_{MB1/P2/ROxS1/MB2/RS}$	200
$T_{S1/MB2/RSxAS}$	3
$T_{ClientxS1/MB2/RS}$	3
$T_{ClientAS}$	3
$T_{S2/RPClient}$	200

$$1\times T_{P1}+3\times T_{MB1/P2/RO}+2\times T_{AS}+5\times T_{S1/MB2/RS}+3\times T_{MB1/P2/RO\times S1/MB2/RS}+2\times T_{S1/MB2/RS\times AS}=1147\text{ ms} \quad (3)$$

$$1\times T_{P1}+1\times T_{MB112/RO}+2\times T_{AS}+2\times T_{S1/MB/RS}+1\times T_{S2/RP}+2\times T_{Client}+1\times T_{P1\times MB1/P2/RO}+1\times T_{MB1/P2/RO\times S1/MB2/RS}+2\times T_{S1/MB2/RS\times AS}+2\times T_{client\times S1/MB2/RS}=578\text{ ms} \quad (4)$$

$$2\times T_{S1/MB2/RS}+2\times T_{Client}+2\times T_{Client\times S1/MB2/RS}=92\text{ ms} \quad (5)$$

Equation (1) shows that 1207 ms is required for protection and authorization. For initial access, Equation (2) shows that 164 ms is required. Equation (3) shows the time of initial publishing is 1147 ms. For subsequent publishing, the time required is 578 ms as shown by Equation (4).

The following equations are also used in the evaluation:

$$\rho=\lambda/\rho \quad (6)$$

$$L_q=\frac{\rho^2}{1-\rho} \quad (7)$$

$$W_q = L_q / \lambda \tag{8}$$

$$W = W_q + 1/\mu \tag{9}$$

$$L = \lambda W \tag{10}$$

where L is the average number of messages in the system. L_q is average number of messaged in the queue of MQTT broker. W is the average service time of a messege in the system. W_q is the average service time of a message in a MQTT broker. P is the utilization of the system. λ is the arrival time.μ is the service rate.

In Figure 7, the function *Access* (*R*, *RPT*) is invoked only when the client initiates the request; otherwise, the published message reaches to the *RP* as a subscriber to the subject. Equation (5) shows the subscriber transaction time is 92 ms.

6. Results and Discussion

6.1. Results

The system is M/M/1 queuing model with exponential distribution of inter-arrival time and service time. The system is tested with a mean inter-arrival time between 588 ms and 640 ms. The mean service rate is *mu* = 1/587, which is the time required for publishing.

The system utilization is computed by Equation (6) and shown in Figure 9.

Mean waiting number to be served is computed by Equation (7) and shown in Figure 10.

Mean waiting time is computed by Equation (8) and shown in Figure 11. Mean waiting number verses mean waiting time is shown in Figure 12, they show a linear relationship indicating that they grow at the same rate.

Mean waiting time in the system is computed by Equation (9) and shown in Figure 13.

Mean waiting number in the system is computed by Equation (10) and shown in Figure 14. The proportion of time the server is idle is computed by Equation (11) and shown in Figure 15.

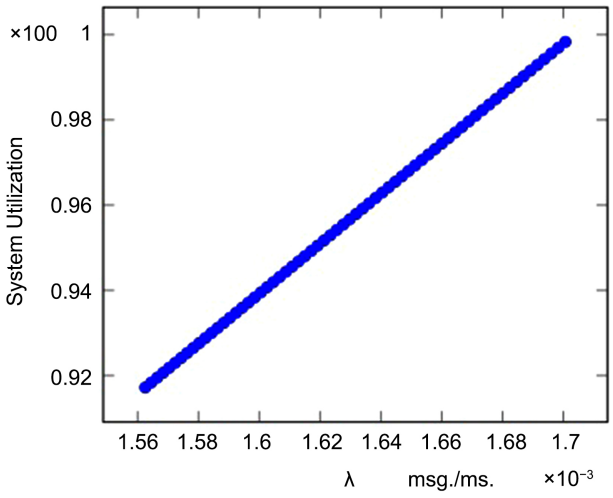

Figure 9. System utilization.

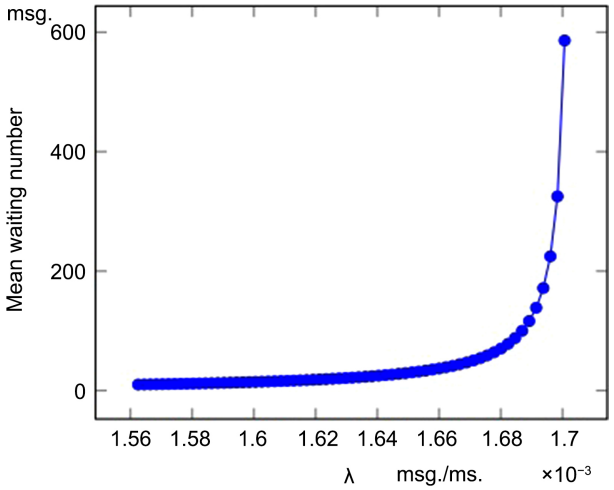

Figure 10. Mean waiting number.

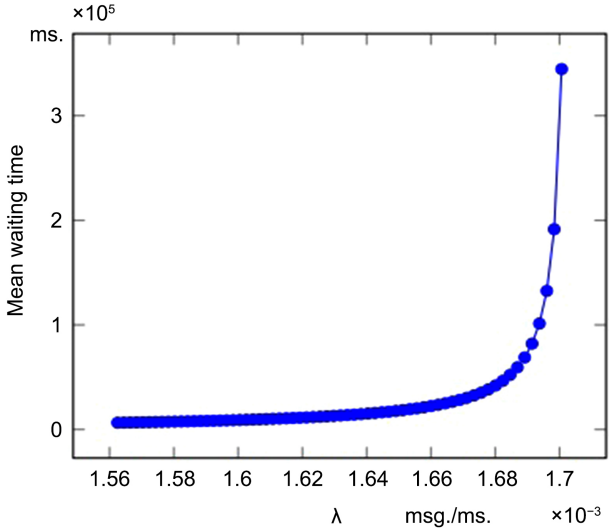

Figure 11. Mean waiting time.

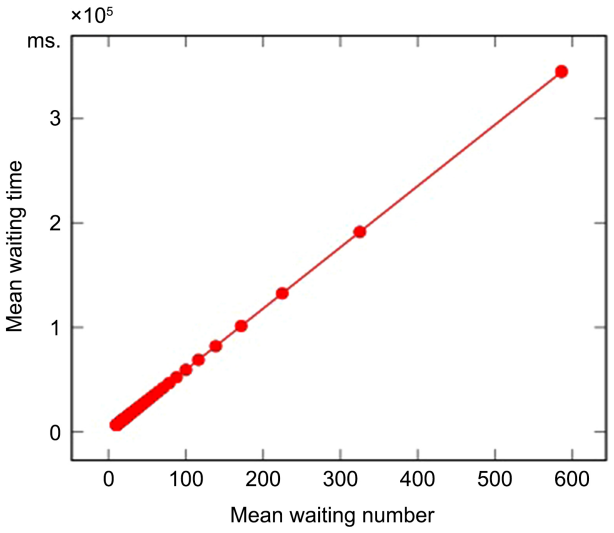

Figure 12. Mean waiting time vs. mean waiting number.

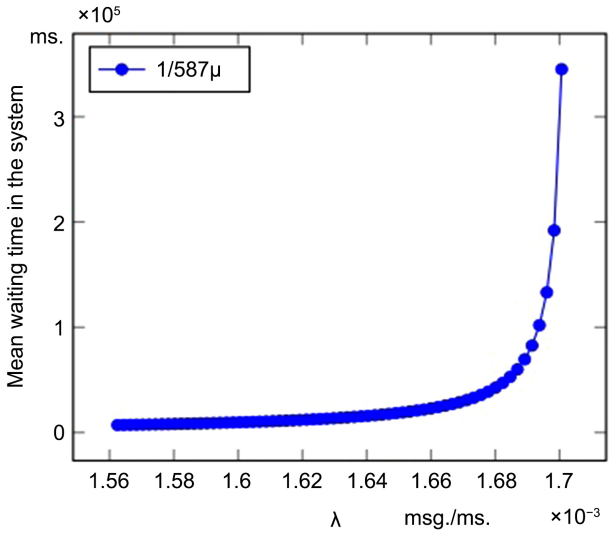

Figure 13. Mean waiting time in the system.

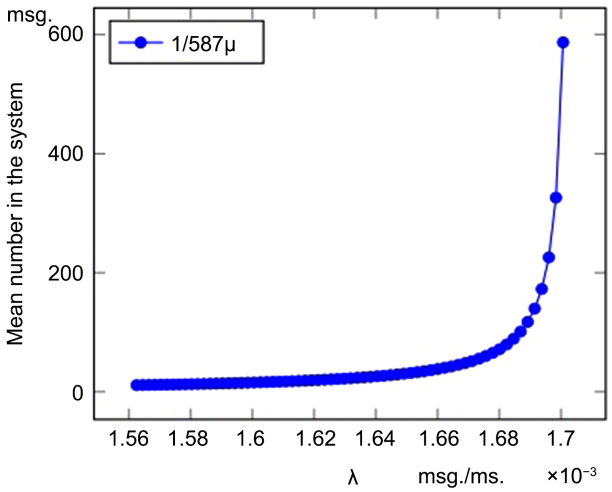

Figure 14. Mean number to the system.

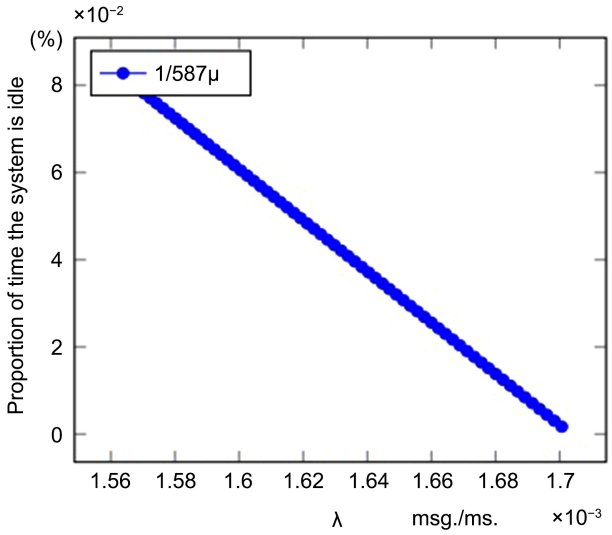

Figure 15. Proportion of time the system is idle.

6.2. Discussion

As the experiments in Sections 5.2 and the results are shown in 6.1 the overhead created between UMA and MQTT is the main critical overhead by the mode. The UMA and MQTT are originally separate models, each model with different features and objectives.

In the proposed model, MQTT and UMA are integrated to provide an extra layer for IoT devices of security. The main point of adding UMA to MQTT or the vice versa is to add accessibility for users for IoT devices.

Any IoT device is not connected to the world unless connected to a broker or a type of protocol that allows the IoT device to send and receive data. However, the accessibility of IoT devices is expected to be quite high; therefore, UMA provides high scaleability for IoT device to publish its data to the wide range of users or *RP*. *RP* could be another IoT device.

Nowadays with the expected future and developing project of smart cities and buildings, IoT devices are expected to send a very large amount of data as (*i.e.* Big Data). Also, the integration of plug and play is a must future for IoT. Now, UMA will establish protocols as well as MQTT. Adding the two models should be done without sacrificing performance.

If there is a trade-off, the system administrator and project developer can discuss the QoS of any model of expected growing IoT projects.

For MQTT, the model is added to UMA to add the subscriber functionality. The model shown in **Figure 6**; has mainly $S1$ and $MB2$ added to the *RS* and $S2$ added to *RP*.

Effectively, $P1$, $MB1$ and $P2$ are not considered an overhead since it is external processing overhead to the UMA protocol. Therefore, the overhead is with $S1$, $MB2$ and $S2$. When the data are received from $P1$ to $MB1$, the data are published to $S1$. $S1$ is represented by $MB2$, which is represented by *RS*. The integration between $MB2$ and *RS* is done by considering any newly published data received by $MB2$ as a request received by an *RP* which is registered as a subscriber for a specific topic. Hence, the time required to get a message about a new topic is less than the time required for a request by *RP*, *Client*, $S2/RP/xClient$, $ClientxS1/MB2.RS$ and $S1/MB2/RS = 346$ compared to the time required for the publishing as mentioned earlier by Equation (4). The time required is only 26% of the time required if the functionality is done by UMA only. Consequently, the performance gain is 74% with MQTT for the publishing function of a new topic.

7. Conclusion

IoT environment is promising for future applications in different domains, such as smart cities or the sensing as a service (SenaaS) model of business. When an IoT application works it is expected that the number of devices grows exponentially as well as the size of data. Security is required to secure access to the data sources; hence, data need to be protected without compromising performance. Therefore, an integrated security layer is required. In this work, the UMA secu-

rity protocol is used to add security with the well-known IoT application protocol MQTT. There are different IoT application protocols; however, MQTT is the most popular for implementation simplicity. This work has shown how to integrate MQTT and UMA in one fast performance and secure model. The integration is maintained by mapping the functionality of both protocols. Performance comparison has shown that there is performance gain for different functionalities; mainly, the publishing function which is the main function of any IoT device. The model has successfully connected isolated IoT devices to the publishing network without compromising security. Moreover, data are published and accessed without accessing the source of the data. Therefore, this hybrid model presented shows a viable potential for secure high-performance Internet of Things.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] ISO/IEC 20922:2016 MQTT v3.1.1. https://www.iso.org/standard/69466.html.

[2] User Managed Access (2015) Kantara Initiative Staff.

[3] Biswas, A.R. and Giaffreda, R. (2014) IoT and Cloud Convergence: Opportunities and Challenges. 2014 *IEEE World Forum on Internet of Things* (*WF-IoT*), Seoul, 6-8 March 2014, 375-376. https://doi.org/10.1109/WF-IoT.2014.6803194

[4] Zhao, F. (2010) Sensors Meet the Cloud: Planetary-Scale Distributed Sensing and Decision Making. 9*th IEEE International Conference on Cognitive Informatics* (*ICCI*'10), Beijing, 7-9 July 2010, 998-998. https://doi.org/10.1109/COGINF.2010.5599715

[5] Alam, S., Chowdhury, M.M.R. and Noll, J. (2010) Senaas: An Event-Driven Sensor Virtualization Approach for Internet of Things Cloud. 2010 *IEEE International Conference on Networked Embedded Systems for Enterprise Applications*, Suzhou, 25-26 November 2010, 1-6. https://doi.org/10.1109/NESEA.2010.5678060

[6] Nikolov, N. and Nakov, O. (2019) Research of Secure Communication of ESP32 Iot Embedded System to .NET Core Cloud Structure Using MQTT SSL/TLS. 2019 *IEEE XXVIII International Scientific Conference Electronics* (*ET*), Sozopol, 12-14 September 2019, 1-4. https://doi.org/10.1109/ET.2019.8878636

[7] Singh, M., Rajan, M.A., Shivraj, V.L. and Balamuralidhar, P. (2015) Secure MQTT for Internet of Things (IoT). 2015 *Fifth International Conference on Communication Systems and Network Technologies*, Gwalior, 4-6 April 2015, 746-751. https://doi.org/10.1109/CSNT.2015.16

[8] Wang, X., Zhang, J.E., Schooler, M. and Ion, M. (2014) Performance Evaluation of Attribute-Based Encryption: Toward Data Privacy in the IoT. 2014 *IEEE International Conference on Communications* (*ICC*), Sydney, 10-14 June 2014, 725-730. https://doi.org/10.1109/ICC.2014.6883405

[9] Perera, C., Zaslavsky, A., Christen, P. and Georgakopoulos, D. (2014) Context Aware Computing for the Internet of Things: A Survey. *IEEE Communications Surveys Tutorials*, **16**, 414-454. https://doi.org/10.1109/SURV.2013.042313.00197

[10] Dai, H., Zheng, Z. and Zhang, Y. (2019) Blockchain for Internet of Things: A Survey. *IEEE Internet of Things Journal*, **6**, 8076-8094. https://doi.org/10.1109/JIOT.2019.2920987

[11] Niruntasukrat, A., Issariyapat, C., Pongpaibool, P., Meesublak, K., Aiumsupucgul, P. and Panya, A. (2016) Authorization Mechanism for MQTT-Based Internet of Things. 2016 *IEEE International Conference on Communications Workshops* (*ICC*), Kuala Lumpur, 23-27 May 2016, 290-295. https://doi.org/10.1109/ICCW.2016.7503802

[12] Ullah, R., Ahmed, S.H. and Kim, B. (2018) Information-Centric Networking with Edge Computing for IoT: Research Challenges and Future Directions. *IEEE Access*, **6**, 73465-73488. https://doi.org/10.1109/ACCESS.2018.2884536

[13] Alhazmi, O.H. and Aloufi, K.S. (2019) Fog-Based Internet of Things: A Security Scheme. 2019 2*nd International Conference on Computer Applications Information Security* (*ICCAIS*), Riyadh, 1-3 May 2019, 1-6. https://doi.org/10.1109/CAIS.2019.8769506

[14] Aloufi, K.S. and Alhazmi, O.H. (2020) Secure IoT Resources with Access Control over Restful Web Services. *Jordanian Journal of Electrical Engineering*, **6**, 63-77. https://doi.org/10.5455/jjee.204-1581015531

[15] Cruz-Piris, L., Rivera, D., Mersa-Maestre, I., De la Hoz, E. and Velasco, J.R. (2018) Access Control Mechanism for IoT Environments Based on Modelling Communication Procedures as Resources. *Sensors*, **18**, 917. https://doi.org/10.3390/s18030917

[16] Aloufi, K. (2019) 6loWPAN Stack Model Configuration for IoT Streaming Data Transmission over CoAP. *International Journal of Communication Networks and Information Security*, **11**, 304-3012.

[17] Yokotani, T. and Sasaki, Y. (2016) Transfer Protocols of Tiny Data Blocks in IoT and Their Performance Evaluation. 2016 *IEEE* 3*rd World Forum on Internet of Things* (*WF-IoT*), Reston, VA, 12-14 December 2016, 54-57. https://doi.org/10.1109/WF-IoT.2016.7845442

[18] Yokotani T. and Sasaki, Y. (2016) Comparison with HTTP and MQTT on Required Network Resources for IoT. 2016 *International Conference on Control*, *Electronics*, *Renewable Energy and Communications* (*ICCEREC*), Bandung, 13-15 September 2016, 1-6. https://doi.org/10.1109/ICCEREC.2016.7814989

[19] Hardjono, T., Maler, E., Machulak, M. and Catalano, D. (2013) User-Managed Access (UMA) Profile of OAuth 2.0 Internet-Draft Draft-Hardjono-Oauth-Umacore-13. https://tools.ietf.org/html/draft-hardjono-oauth-umacore-13

[20] Jones, M. and Hardt, D. (2012) The OAuth 2.0 Authorization Framework: Bearer Token Usage. https://tools.ietf.org/html/rfc6750

[21] Siriwardena, P. (2014) Advanced API Security: Securing APIs with OAuth 2.0, OpenID Connect, JWS, and JWE. Apress, Pittsburg. https://doi.org/10.1007/978-1-4302-6817-8

[22] Maler, E., Machulak, M., Richer, J. and Hardjono, T. (2019) User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization. Internet-Draft Draft-Maler-Oauth-Umagrant-00, Internet Engineering Task Force.

[23] Al-khafajiy, M., Baker, T., Waraich, A., Al-Jumeily, D. and Hussain, A. (2018) Iot-Fog Optimal Workload via Fog Offloading. 2018 *IEEE/ACM International Conference on Utility and Cloud Computing Companion* (*UCC Companion*), Zurich, 17-20 December 2018, 359-364. https://doi.org/10.1109/UCC-Companion.2018.00081

[24] Jaloudi, S. (2019) Mqtt for Iot-Based Applications in Smart Cities. *Palestinian Journal of Technology and Applied Sciences*, **2**, 1-13.

[25] Osipov, M. (2008) Home Automation with Zigbee. In: Balandin, S., Moltchanov, D. and Koucheryavy, Y., Eds., *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, Springer, Berlin, 263-270. https://doi.org/10.1007/978-3-540-85500-2_26