A Brief Review on MQTT's Security Issues within the Internet of Things (IoT)

Ahmed J. Hintaw[1], Selvakumar Manickam[1], Shankar Karuppayah[1], and Mohammed Faiz Aboalmaaly[2]
[1] National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Penang, Malaysia
[2] Department of Computer Techniques Engineering, Alsafwa University College, Iraq
Email: {aj.hintaw, selva, shankar}@nav6.usm.my, mohammadfaiz2003@gmail.com

Abstract —The domain of Internet-enabled devices and the associated communication technology is currently experiencing a rapid revolution which blossomed into the paradigm of the Internet of Things (IoT). IoT offers a number of innovation capabilities and features, but they are also prone to security vulnerabilities and risks. These vulnerabilities must be studied to protect these technologies from being exploited by others. Message Queuing Telemetry Transport (MQTT) is an application layer protocol that is vulnerable to various known and unknown security issues. This review paper intends to investigate and classify the available security methods that are commonly used as a security solution and highlight the weaknesses of the current proposals. In a nutshell, the following are reviewed: (i) the concepts of MQTT protocol in IoT, (ii) common security solutions in MQTT protocol in IoT, (iii) security levels in MQTT protocol. Finally, this review presents a set of guidelines for researchers to choose the right security mechanism for different applications in IoT.

Index Terms—MQTT, confidentiality, internet of things, publish-subscribe, security

I. Introduction

Internet protocol IPV6 and digital revolution allows employing IoT quickly and everywhere. In the next five years, trillions of IoT gadgets are going to be used in the IoT domain[1]. Identity theft, data integrity of devices, and device-to-device communications, were not strictly addressed. Further, most of the proposed security solutions and their associated privacy aspects are still at an infancy stage [2]. Cryptography techniques and approaches such as public key infrastructure, identity-based encryption; are commonly used to address and deal with security vulnerabilities. These are proposed to secure communication of IoT [1]-[3]. The current techniques do not deal at the protocol level; it simply attempts to resolve at the primitive security level for machine to machine communications.

Existing IoT application layer protocol such as MQTT which is based on TCP [3], [4], has limited security features such as only for simple authorization policies and basic authentication. Hence, security issues of this protocol need to be addressed to adopt it in IoT. Moreover, MQTT is ubiquitous, particularly in the domain of sensor and social networks as well as vehicle to vehicle communications[5]. Therefore, in this paper, the focus is on studying the security of MQTT protocol. TLS/SSL with session key management and certificates is suggested to secure MQTT protocol. However, an effective security mechanism is required to secure MQTT and should be lightweight for adoption in IoT because environments of IoT consist of massive numbers of heterogeneous things. Hence, key exchanges and storing the certificates for each session is very heavy in terms of processing time, energy consumption and memory space. Moreover, various attacks such as Heartbleed, CRIME, BEAST, RC4, that exploits TLS/SSL, has widely affected the IoT devices throughout the globe.

This paper will also review the security mechanisms to secure MQTT protocol in IoT. Researchers have a clear comprehension of various MQTT security techniques in the environment of IoT. The remainder part of this paper is organized as follows: Section II presents the overview of MQTT. Section III describes the MQTT security mechanisms available from the literature. Security levels of MQTT in IoT are presented in Section IV; the drawbacks of the existing security mechanisms for MQTT protocol in IoT are discussed in Section V. Finally, Section VI draws the study's conclusion.

II. Overview of MQ Telemetry Transport

MQTT is a lightweight message-oriented protocol and widely implemented based on a publish-subscribe programming model; thus, decreasing the drain of IoT resources. The main goal of MQTT is to use minimal network bandwidth than HTTP or similar protocols. Moreover, it has been used in numerous fields where the low throughput provided by communication links such as monitoring of SCADA, automation, or vulnerable to low accessibility, like satellite links. The preliminary version of MQTT is 3.1. It has come to be a standard by the Advancing Open Standard for the Information Society (OASIS) and version 3.1.1 refers to the latest release of protocol specifications[4]. Message format of MQTT consists of three portions: a fixed header, shown in Table I; a variable header; and a payload. From the standpoint of protocol, MQTT is based on TCP, giving the control of error and flow to the single packets of the protocol stack lower layers. MQTT is implementing a publish-subscribe programming model, as declared above; this implies that:

Manuscript received August 12, 2018; revised April 25, 2019.
Corresponding author email: aj.hintaw@nav6.usm.my.
doi:10.12720/jcm.14.6.463-469

TABLE I: MQTT HEADER

Bit->	7	6	5	4	3	2	1	0
Byte 1	Msg type				DUP	QoS		Retain
Byte 2	Remaining length							
	Variable Header							
	Payload							

Topics: Classifies the information. Topic structures have no order or rules, but hierarchies are permitted via separators with the form of file system path called topic levels e.g. "office/light/status".

Clients or nodes (getting access to the published data): Nodes can get access to acts as reading action to the published date in a particular topic via subscribing to the similar topic. Clients should publish data under specific topics and can subscribe published data to other corresponding entities accessed to the same broker or server via that given topic. The procedure of subscription is achieved by relaying a SUBSCRIBE message command to the server or broker that will be responsible to join the requesting node requiring access to that topic.

Nodes can generate content that acts as writing action: By publishing information to a specific topic, the published data procedure is achieved by relaying a PUBLISH message command to the broker that will not be responsible for any treatment on the message transmitted. It sends the payload message to all nodes that earlier subscribed to the same topic. MQTT acts as one of the worldwide used protocols in different domains as shown in Table II.

TABLE II: SOME SOLUTIONS THAT USE MQTT PROTOCOL

Smart home	Brokers	Clients
Homegear	Mosquitto	CocoaMQTT
Domoticz	ActiveMQ	emqttc
Lelylan	hbmqtt	mqtt-client
cul2mqtt	HiveMQ	M2Mqtt
aqara-mqtt	Moquette	mqtt cpp
Home.Pi	Mosca	mqttex
Home Assistant	VerneMQ	Paho
pimatic	hrotti	rumqtt
FHEM	SurgeMQ	hbmqtt

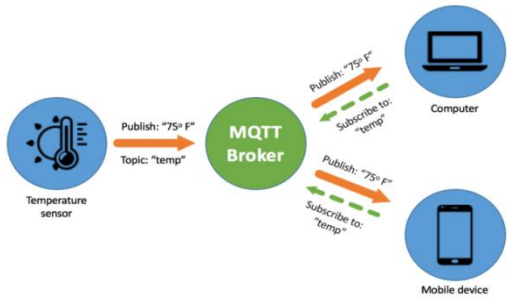

Fig. 1. Publish-Subscribe paradigm in MQTT system

Fig. 1 describes the publish-subscribe paradigm in MQTT system. In particular, MQTT broker is the only entity that has complete information of the network structure itself. Connection of a client to client is not possible. Furthermore, it requires two separate moments for dispatching and reception of messages. For instance, sometimes a node is not available; therefore the dispatching process will be delayed until the node becomes available.

III. CURRENT SECURITY SOLUTIONS

The research communities have given a lot of attention to the privacy and security issues in IoT and they have handled these issues to present solutions at different levels. A number of security techniques to secure application layer protocols in IoT has been proposed and among of them is MQTT. It is one of the most commonly adopted protocols in IoT. Most of the security problems are related to the state in which the protocol works by default. This section presents the previous works that stand as background for the study. It presents the works where the implementation of different cryptographic algorithms is used for comparative analysis and creating new security solutions to secure MQTT protocol in IoT. Fig. 2 shows the current security solutions diagram.

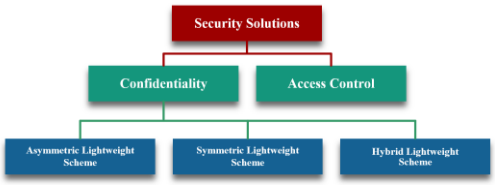

Fig. 2. The Diagram of the current security solutions

A. Confidentiality

The authors in [6] explains data confidentiality acts as a fundamental issue for IoT solutions, especially those related to the business area. The existing solutions for data confidentiality may not be suitable according to two main challenges: the size of generated data and the effectiveness in the data access control of dynamic data flows. The authors mentioned that data confidentiality can be achieved by using the right identity management. The cryptography schemes can achieve the confidentiality of the communications channel. Based on the capability, mission critical and the application of the IoT system, the existing asymmetric and symmetric algorithms must be studied before implementing them as listed by [7]. Numerous methods are depicted by the research community to present the confidentiality of transmitted data.

1) Asymmetric lightweight scheme

To enhance security for MQTT, in [8] the authors have indicated that it is not possible in all IoT scenarios to use TLS\SSL with session key management and certificates and, it is not viable. In particular, a secure version of MQTT has been proposed, their solution is to adopt lightweight attribute-based encryption (ABE) [8], [9] using elliptic curves [1]. They have illustrated the feasibility of their proposed solution for different IoT demands.

2) Symmetric lightweight scheme

An overview of cryptography in embedded systems proposed by [10]. They consider the basic concepts,

characteristics, and goals of various cryptographic algorithms. [11] It deals with resources limitation challenges of smart devices to implement asymmetric schemes to perform authentication process, and to offer an authentication approach that can fit the resource limitations based on methods such as hash functions or "OR" operations. In [12] authors have proposed a mechanism to secure application layer in IoT by using dynamic key cryptography XOR operation. Lightweight encryption algorithms - Secure IoT (SIT) has been proposed by [13], where the developed algorithm is restricted to a 64-bit key size and blocks cipher, used hybrid methods to get advantages of Feistel structure and SP (substitution-permutations) network to make it lightweight and cryptographically secure to be accepted in IoT environments, but it needed to be tested in different software and hardware for potential attacks.

Authors in [14] have highlighted the most applicable mechanisms in the area of the industrial IoT by presenting a comprehensive study of several MQTT security, the first evaluation option is to use AES-CCM to implement encryption of Link layer to achieve hop-by-hop protection. In this scenario, LLSEC driver is the appropriate option to activate good security interactions between sensor and broker by ignoring the weakness of its "hope by hope" process to use instead of that single hope. The second evaluation indicates, offering "end to end" option via using AES, AES-OCB, AES-CBC to encrypt the payload. It is an attractive option to encrypt the payload with AES-OCB: it adds more security compared to AES-CCM and the AES-OCB option could not be handled when the payload amount is 64-byte, they have focused on the wind park scenario for their evaluation to impose it in the area of actual industrial use.

3) Hybrid lightweight scheme

The architecture of hybrid technique targets joining different approaches to use the advantages of both asymmetric and symmetric schemes. Hybrid schemes target particular applications such as RFID tags, Internet servers, and movable devices. The author in [15] proposed a new hybrid scheme, which allows using smaller sized key compared to existing cryptographic solutions by comparing different asymmetric (RSA ,ECC) and symmetric algorithms (AES 128, XTEA, HIGHT, RC5 and PRESENT) to investigate which lightweight algorithm is better to implement, and modify this scheme due to the data sending scenario, the hybrid scheme includes two types of cryptography at the same time: symmetric algorithms for encryption/decryption data; and asymmetric algorithms for key exchange.

Authors in [3], [16] have used ABE in order to encrypt the private key of symmetric algorithm AES to provide broadcast encryption by implementing both KP\ABE[8] and CP\ABE[9] and using AES to encrypt the message. This scheme achieved message confidentiality and broadcast encryptions. Authors in [2] have used the ABE scheme provided in [3] to ensure the security of publish-subscribe (Pub-Sub) architecture based IoT, and they used various factors such as memory space, processor usage and execution time to evaluate the scheme's various security levels.

Authors in [17] have contributed to present message confidentiality and access control by implementing AES dynamic S-box with ABE, where secret key of AES encrypted by ABE and MQTT payload is encrypted by AES, authors deal with KP-ABE to make it lightweight [8] over lightweight ECC [1].

B. Access Control

Other approaches have been exhibited by some authors. Moreover, some researchers have tried to deal with the general problems of IP-based protocols used by IoT devices, one of which is MQTT. In these cases, the authors focus on the security of this type of device as part of a broader spectrum, treating the layers of protection that can wrap around the TCP/IP protocol and the security architecture and models that best fit IoT networks [18].

An interesting approach *SecKit* is a model-based security toolkit that tries to force the use of a series of security policies so that the protocol implements some protection measures that are not found in its default implementation [19]. The authors in [20] proposed a methodology to permit dynamic enforcement of usage control (UCON) in MQTT architecture and workflow, without requiring protocol modifications. The proposed UCON methodology improved the security level of the MQTT protocol. New policies and key management framework to secure MQTT presented by [21] called Authenticated Publish Subscribe, the mechanism introduces a secure publish-subscribe scheme to protect MQTT.

IV. Security Levels of MQTT in IoT

A. Physical/Perception Layer

Running the systems on private networks was the one inherent security solution of MQTT at the beginning of industrial usage. Nowadays, the same security level can be utilized as a standard MQTT. Several scientists recommend using on top of other solutions peer-to-peer structured network [22], whereas others proposed techniques based on nodes forming a point-to-point network, hence limiting the shared communication channels from those hazards [23]. In these situations, security is obtained by segregating physical units, but may not be practicable in some scenarios where sensors are placed in unrestricted areas or extremely hazardous if the installation domain is not quite secure.

B. Network Layer

The preferred option to achieve confidentiality, as well as authentication, is the IPSec at the network layer. However, due to the extra headers, it causes an overload. Such issues can be solved by implementing 6LowPAN compression and the usage of strong cryptographic schemes. The Host Identity Protocol is another applicable

option at the network field where identifiers of broker and nodes could be decoupled from their locators. Within this situation, the benefits may move into host mobility as well as multi-homing, but the disadvantage is difficulties of employing the cryptographic schemes as well as in the public key distribution.

C. Transport Layer

Since most of MQTT devices suffer from resource limitations as well as due to the extra workload needed to set secured connections and to encrypt all traffic, it's not possible to implement TLS in these devices. In such situations, TLS Session Resumption capability can be used, significantly decreasing the repetition of all TLS Handshake process. To reduce the issues regarding the computational load because of the ciphering approaches performed [24], this could be solved by using a hardware-based TLS implementation, but a further element requires to be installed and overall device cost and power consumption will increase. The utilization of TLS and certificates, whenever possible is recommended by OASIS, to have confidentiality, integrity, and authentication. In [25], the first use of TLS for IoT applications lists the vulnerabilities. These are essentially regarding:

Configuration: The protocol security can significantly be lower and could be exposed to attacks because the TLS layer configuration is improper or invalid cipher suites or limited utilization.

TLS vulnerabilities: list popular attacks that have been used against TLS [26]. Most of these threats are associated with vulnerable cipher recommended in an earlier edition of TLS or SSL or maintained back-compatibility with an earlier edition of the protocol.

Certificates management: the management of the certificates is a major concern among most researchers; it is an intractable job to perform such revoking, updating or managing the certificates on scattered devices when completing the initial deployment. The remote devices that use certificates have common issues regarding revoking and updating. In fact, it's a tough task to perform frequent cipher updating and configuration over unreliable links on a huge number of nodes, and performing analysis based on the parameters such as energy consumption and throughput as extra traffic must be considered. In addition, TLS utilization can be implemented at commercial and open-source solutions of all MQTT broker to ensure authentication and confidentiality of data in all nodes.

D. Application Layer

Within the application layer, an external authentication system to the broker itself can be used in case of multiple-items connecting systems. Authentication systems such as OAuth or LDAP are possible to be implemented on the MQTT standard for these applications. In both s`1ituations, based on given credentials, allowing access to tokens and authentication will be the responsibility of the external system. An extra layer of complexity surely will be introduced when implementing such systems, and during the design stage, pros and cons must be accurately weighted. Moreover, implementing OAuth and LDAP on top of a TLS connection can cause potential problems as highlighted earlier. The protocol stack and the security mechanism available at each layer are presented in Table III.

TABLE III: MQTT SECURITY MECHANISM AT EACH LAYER

Layer	Application	Transport	Network	Perception
Protocol used	MQTT	TCP	IPv6,RPL	IEEE 802.15.4 PHY, MAC
Security protocol	Not fixed	TLS	IPSec	IEEE 802.15.4 security
Confidentiality	No	Yes	Yes	Yes
Integrity	No	Yes	Yes	Yes
Authentication	Not fixed	Broker	Not fixed	Node
To be secured	Publisher to subscriber	Publisher to subscriber	Node -to- node	Air interface

V. DISCUSSION

This section aims to provide a guideline for researchers to create new security solutions to secure MQTT protocol in IoT. Different criteria are used to benchmark the differences between the existing security mechanisms. Such criteria include execution time, memory usage and CPU as well as energy consumption under different security levels. Due to node resource limitations in some IoT scenarios, deploying high computational cryptography authentication schemes will be problematic. So, such situations that relate to resource limitations can be solved by implementing lightweight cryptography that is suitable in IoT. Various security options mentioned in literature such as lightweight symmetric and asymmetric schemes e.g ABE over elliptic curves, SIT, and AES were presented. The current security options do not present a satisfactory security level. Integrity and confidentiality can be presented by the use of the symmetric scheme, but it does not provide authentication features that could affect availability, whereas the asymmetric scheme has a large key size, hence that makes the scheme quite slow. Existing mechanisms for MQTT protocol have some drawbacks and these drawbacks can be categorized into three main categories which are discussed below and shown in Fig. 3:

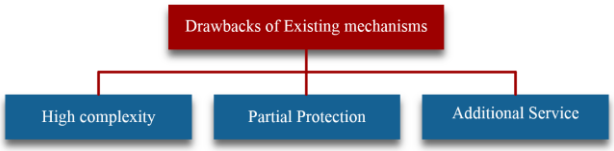

Fig. 3. Drawbacks of Existing mechanisms

A. High Complexity

Through operations, execution of any mechanism can reveal the complexity of that mechanism [27]. Therefore, when the mechanism is more complex it means that it consumes more resources to execute the operation i.e MQTT payload encryption-decryption process in IoT. From the study [28], it was found that the existing mechanisms such as traditional public key schemes have high complexity issue due to their structural design when implementing message encryption-decryption during the publish-subscribe process in devices. In addition to that, confidentiality, integrity, and authenticity can be achieved by implementing the asymmetric scheme, but such schemes will cause an energy drain from the IoT nodes due to the large key size which makes the schemes more complex. Furthermore, security in MQTT depends on the use case and selection of broker [25]. Most brokers provide security based on TLS, but TLS affects the performance significantly, especially CPU usage.

B. Partial Protection for MQTT Protocol

Although some of the existing mechanisms such as [29] have addressed the complexity issues, a security protocol for IoT has been designed in which a specific master key is imprinted in the devices which are either fixed or static. Then a challenge-based shuffling algorithm is used at the client and server side to make the key dynamic. Nevertheless, studies [12] have shown that these mechanisms have introduced new vulnerabilities. The question here is how client and server can choose the same challenge for shuffling the key when they are distant/unknown to each other. Further, the dynamic session key is left exposed for a reasonable amount of time during which an attacker can easily decipher the messages and this can compromise the entire network. Therefore, an attacker can exploit these mechanisms and can easily decipher the publish-subscribe MQTT protocol messages.

C. Additional Service Requirements

Some of the mechanisms were introduced i.e. traditional encryption schemes to ensure message confidentiality of MQTT by encrypting MQTT payload at the application layer. Therefore, it's viable to use any available encryption scheme if target devices have adequate resources. However, in [2] the amount of data produced continuously increases due to the significant growth in the huge number of connected devices in IoT. Collected data (e.g., behavior patterns, presence) in a particular sector are vulnerable to attackers and can be diverted away from the original destination. Therefore, privacy is of crucial importance. However, techniques such as traditional preserving privacy are limited in a number of directions: flexible data sharing, fine-grain access control, key management scalability.

Due to these constraints the implementation of the security mechanisms for MQTT protocol is limited. Therefore, new security approaches are required to secure MQTT publish-subscribe messages in IoT.

VI. Conclusion

MQTT is one of the protocols that is standardized by OASIS and widely used within the IoT ecosystem. Several security solutions and aspects have been discussed in this paper. Traditional public key mechanisms which are used to secure MQTT publish-subscribe messages drains energy resources from the IoT nodes due to the large key size. Therefore, it is impractical to use such mechanisms on devices with low computation resources. Furthermore, security threats of MQTT publish-subscribe messages such as spoofing can cause DDoS attacks on the MQTT broker. However, TLS protocol seems to be a good choice to secure MQTT protocol when the resources of the IoT devices are not limited. In this study, the researchers have shown the drawbacks of the current solutions and specify their limitations within three classifications i.e. partial protection, high complexity, and additional service requirements. Security solution options of MQTT protocol that were presented by the researchers do not provide integrated solutions because most of them have a specific focus. Current solutions need development to be integrated, particularly for the devices that have limited resources.

References

[1] B. S. Adiga, P. Balamuralidhar, M. A. Rajan, R. Shastry, and V. L. Shivraj, "An identity based encryption using elliptic curve cryptography for secure M2M communication," in *Proc. First Int. Conf. Secur. Internet Things - Secur. '12*, pp. 68–74, 2012.

[2] X. Wang, J. Zhang, E. M. Schooler, and M. Ion, "Performance evaluation of Attribute-Based Encryption: Toward data privacy in the IoT," in *Proc. IEEE Int. Conf. Commun.* , 2014, pp. 725–730.

[3] M. Ion, "Security of publish/subscribe systems," University of Trento, 2013.

[4] O. Standard. (2014). MQTT version 3.1. 1. [Online]. Available: http//docs. oasis-open. org/mqtt/mqtt/v3

[5] E. G. Davis, A. Calveras, and I. Demirkol, "Improving packet delivery performance of publish/subscribe protocols in wireless sensor networks," *Sensors*, vol. 13, no. 1, pp. 648–680, 2013.

[6] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[7] M. Katagi and S. Moriai, "Lightweight cryptography for the internet of things," *Sony Corp.*, pp. 7–10, 2008.

[8] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur. - CCS '06*, 2006, p. 89.

[9] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symposium on Security and Privacy*, 2007, pp. 321–334.
[10] T. Wollinger, J. Guajardo, and C. Paar, "Cryptography in embedded systems: An overview," in *Proc. Embedded World 2003 Exhibition and Conference*, 2003, pp. 735–744.
[11] A. Esfahani, *et al.*, "A lightweight authentication mechanism for M2M communications in industrial IoT environment," *IEEE Internet Things J.*, 2017.
[12] A. Bashir and A. H. Mir, "Securing publish-subscribe services with dynamic security protocol in MQTT enabled internet of things," *Int. J. Secur. ITS Appl.*, vol. 11, no. 11, pp. 53–65, 2017.
[13] M. Usman, I. Ahmed, M. I. Aslam, S. Khan, and U. A. Shah, "Sit: A lightweight encryption algorithm for secure internet of things," arXiv Prepr. arXiv1704.08688, 2017.
[14] S. Katsikeas *et al.*, "Lightweight & secure industrial IoT communications via the MQ telemetry transport protocol," in *Proc. IEEE Symposium on Computers and Communications*, 2017, pp. 1193–1200.
[15] O. Khomlyak, "An investigation of lightweight cryptography and using the key derivation function for a hybrid scheme for security in IoT," 2017.
[16] M. Green, S. Hohenberger, B. Waters, and others, "Outsourcing the decryption of abe ciphertexts," in *Proc. USENIX Security Symposium*, 2011, vol. 2011, no. 3.
[17] L. Bisne and M. Parmar, "Composite secure MQTT for internet of things using ABE and dynamic S-box AES," in *Innovations in Power and Advanced Computing Technologies*, 2017, pp. 1–5.
[18] T. Heer, O. Garcia-Morchon, R. Hummen, S. L. Keoh, S. S. Kumar, and K. Wehrle, "Security challenges in the IP-based internet of things," *Wirel. Pers. Commun.*, vol. 61, no. 3, pp. 527–542, 2011.
[19] R. Neisse, G. Steri, and G. Baldini, "Enforcement of security policy rules for the internet of things," in *Proc. IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications*, 2014, pp. 165–172.
[20] A. La Marra, F. Martinelli, P. Mori, A. Rizos, and A. Saracino, "Improving MQTT by Inclusion of Usage Control," in *Proc. International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, 2017, pp. 545–560.
[21] A. Rizzardi, S. Sicari, D. Miorandi, and A. Coen-Porisini, "AUPS: An open source a Uthenticated Publish/Subscribe system for the internet of things," *Inf. Syst.*, vol. 62, pp. 29–41, 2016.
[22] R. H. Weber, "Internet of Things--New security and privacy challenges," *Comput. law Secur. Rev.*, vol. 26, no. 1, pp. 23–30, 2010.
[23] J. L. Espinosa-Aranda, N. Vallez, C. Sanchez-Bueno, D. Aguado-Araujo, G. Bueno, and O. Deniz, "Pulga, a tiny open-source MQTT broker for flexible and secure IoT deployments," in *Proc. IEEE Conference on Communications and Network Security*, 2015, pp. 690–694.
[24] C. Lesjak, *et al.*, "Securing smart maintenance services: hardware-security and TLS for MQTT," in *Proc. IEEE 13th International Conference on Industrial Informatics (INDIN)*, 2015, pp. 1243–1250.
[25] M. Singh, M. A. Rajan, V. L. Shivraj, and P. Balamuralidhar, "Secure mqtt for internet of things (Iot)," in *Proc. Fifth International Conference on Communication Systems and Network Technologies*, 2015, pp. 746–751.
[26] Y. Sheffer, R. Holz, and P. Saint-Andre, "Summarizing known attacks on transport layer security (tls) and datagram tls (dtls)," 2015.
[27] R. L. Flood and E. R. Carson, *Dealing with Complexity: an Introduction to the Theory and Application of Systems Science*, Springer Science & Business Media, 2013.
[28] N. Sklavos and I. D. Zaharakis, "Cryptography and Security in Internet of Things (IoTs): Models, schemes, and implementations," in *Proc. 8th IFIP International Conference on New Technologies, Mobility and Security*, 2016, pp. 1–2.
[29] S. Mishra, "Network security protocol for constrained resource devices in Internet of things," in *Proc. Annual IEEE India Conference (INDICON)*, 2015, pp. 1–6.

Ahmed J. Hintaw was born in Karbala Province, Iraq, in 1986. He received the B.S. degree, Hefei, in 2009 and the M.S. degree from Jamia Hamdard University (JHU), New Delhi, India in 2012, both in computer science. He is currently pursuing the Ph.D. degree with the National Advanced IPv6 Center (NAv6), Universiti Sains Malaysia (USM). His research interests include Internet of Things, Cryptography, and Network Security.

Selvakumar Manickam is the senior lecturer at National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia. He received his Bachelor of Computer Science and Master of Computer Science in 1999 and 2002, respectively. He obtained his Ph.D. from Universiti Sains Malaysia (USM) in 2013. His research interests are Internet security, cloud computing, IoT, Android and open source technology. He is an Executive Council member of Internet Society (ISOC), Malaysian Chapter and also the Head of Internet Security Working Group under Malaysian Research and Education Network (MyREN).

Shankar Karuppayah is currently a Senior Lecturer and researcher at National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia. He obtained his B.Sc Computer Science (USM), Malaysia and the M.Sc. Software Systems Engineering (KMUTNB), Thailand. He obtained his PhD in 2016

from Technische Universität Darmstadt in the field of Cyber Security. His main research interests are P2P Botnets, Distributed Systems and Cyber Security in general. To date, he has authored and co-authored many articles in journals, workshops, and conference proceedings. He is also a reviewer in many esteemed network and security journals.

Mohammed Abomaali, head of Computer Techniques Engineering Department at Alsafwa University College, Iraq. He received a bachelor's degree in software engineering from Mansour University College and a master's as well as a PhD degree in computer sciences from Universiti Sains Malaysia in Penang, Malaysia. His research interests include parallel computing, cloud computing and IoT.