Vulnerabilities and Security Issues in IoT Protocols

Raghad M. Abdulghani*, Marwa M. Alrehili*, Abrar A. Almuhanna*, Omar H. Alhazmi
Department of Computer Science, Taibah University, Medina, Saudi Arabia
raghad.at@gmail.com, mmrehil@gmail.com, abrar_ali_m@hotmail.com, ohhazmi@taibahu.edu.sa

***Abstract*—The spread of the Internet of Things (IoT) has improved our daily life because it blends the physical and the virtual world. Thus, the IoT is device-to-device (D2D) communications that monitor changes in the environment and interacts with them. Also, it can be used from anywhere in the world due to the interconnected nature of the Internet. However, it could violate the privacy of individuals by revealing sensitive information collected by IoT devices about a user's online activities. Also, as the IoT is evolving quickly with new technologies, more privacy issues and abuse of the collected data will arise. Hence, we need to provide more secure communication protocols. In this paper, we will review the security and privacy problems in some IoT protocols. As well as reviewing the proposed solutions to certain problems. Lastly, we will discuss three of the most popular protocols used in the IoT, which are: data distribution service (DDS), message queue telemetry transport (MQTT), and long range wide-area network (LoRaWan).**

***Index Terms*—IoT, Security, Privacy, Protocols, MQTT, DDS, LoRaWan**

I. Introduction

Currently, the Internet is a fundamental utility everyday life, where in the past, using the Internet was mainly for web services, e-commerce and emails. Now, however, we introduce a new smart device every day that connects to the Internet [1]. Take into consideration the Internet of Things (IoT), which is one of the most critical technologies. It can be described as a network of wirelessly connected things, devices, and objects that can sense, act, and interact with each other and with the environment [2].

In IoT architecture, there are three main layers that exist in any system. As shown in Fig. 1, these layers can be specified as the perception layer, network layer, and application layer [3]. Furthermore, the perception layer contains hardware devices that are used to interact with the environment and gather data that are needed. As for the network layer, it is the link between IoT devices and platforms. They are used to transform and send data gathered from devices, and to return a response from the application layer. On the other hand, the application layer is simply a platform that contains a collection of services and tools that are used to help IoT developers to develop and run IoT applications [4].

Since the application layer focuses on bringing people and things together, it provides an end-user interface for IoT applications. Thus, one of the most popular protocols in the application layer is the data distribution service (DDS) protocol. It was proposed by object management group (OMG)

*These authors contributed equally

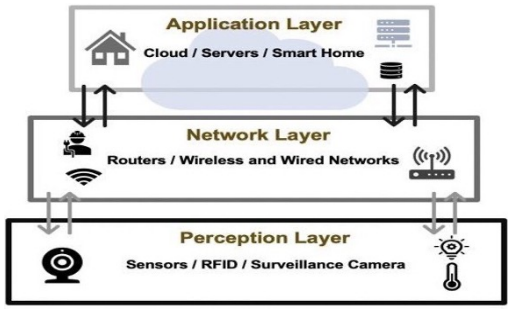

Fig. 1. Three-layered architecture of the IoT [4].

to the IoT system for its ability to provide real-time data distribution for IoT devices [5]. Although it may face some security issues, including eavesdropping, masquerading, and network disruption. As a result, many improvements have been developed for the protocol in order to enhance its security.

Also, message queue telemetric transport (MQTT) is one of the most-used protocols in the application layer, even though there are many application IoT entity protocols and standards, such as constrained application protocol (CoAP). It is a publish-subscribe communication transport that is designed to be applicable for devices with limited resources and low bandwidth usage [6].

In the network layer, one of the most well-known protocols is the long-range wide-area network protocol (LoRaWAN®). The LoRaWAN protocol enables confirmation of the traffic from the terminal to the gateway (uplink) and reverse (downlink), increasing the number of cases that IoT uses that it can aid. But this comes at a time downlink traffic expense has a significant effect on scalability due to one of the duty cycle constraints of a gateway [7].

In this study, we present an overview of some of the most popular IoT protocols DDS, MQTT, and LoRaWAN protocols. Then, we are going to discuss each protocol's structure, some security threats, and their solutions.

The rest of the paper is organized as follows. In Section II, we review some related works on the most popular IoT protocols. In Sections III, IV, and V, we present the structure of DDS, MQTT, and LoRaWAN protocols, with some of their security threats and solutions. In Section VI, we present a comparison between them. The final Section VII concludes the paper with a summary.

978-1-7281-7407-5/20/$31.00 ©2020 IEEE
DOI 10.1109/SMART-TECH49988.2020.00020

II. Related work

Many protocols have been developed for the application layer in order to provide all the information that users need and to connect them to the IoT system. One of these protocols is the DDS protocol. Consequently, DDS is structured in a way that provides a distributed, flexible, and highly secured network. However, it faces security issues, including unauthorized access to data and devices, data manipulation, and disruption of the network [8]. As a result, many improvements have been developed to enhance its security.

Firdous et al. [9] summarized the MQTT protocol threats that are similar to IoT threats, such as malicious internal users, bad manufacturers, and external attackers. They also illustrated many threat scenarios to understand the risk. Further, they tested the broker against two scenarios of denial of service (DoS) attack using the MQTT Mosquitto™ broker. The first scenario was sending large payload MQTT messages in order to exhaust the server resources. For the second attack, they used a transmission control protocol (TCP)-based attack SYN flood in order to exhaust the server bandwidth. As a result, for the first scenario, they advised implementing measures or load-balance in the broker in order to disclose the malicious client and block their messages to save the broker from the flooding. For the second scenario, they suggested that applying an adequate firewall configuration will be a good countermeasure against such an attack, and thus would detect and terminate the malicious connections.

Miller [10] gives a quick overall view of LoRaWAN privacy. And explains how the security mechanisms in the protocol are configured for setting up a LoRaWAN. He explains the position of the main material in a LoRaWAN system and advises that the backend might be compromised by key management flaws. However, the study does not examine the protocol nor determine the security of the exchange of messages.

III. Data Distribution Service Protocol

DDS is a real-time, distributed M2M protocol that supports multicasting technology in order to provide scalability, reliability, and availability [8]. Additionally, it uses a model called publish–subscribe model and another called data centric model. Thus, the publish-subscribe model consists of information providers and information consumers. Information providers publish events to the system, whereas information consumers subscribe to certain events in the system [11]. Moreover, the data centric model considers data to be an essential asset of the system.

A. DDS architecture

Taking a scoped look at the DDS architecture, it consists of two layers: the data centric publish–subscribe layer (DCPS) and the data local reconstruction layer (DLRL). DCPS transmits information to the subscriber, while DLRL is an optional layer that acts as an interface for the functionalities of the DCPS, as shown in Fig. 2 [12].

The DCPS layer is a collection of unrelated data structures that are specified by a topic and a type; the topic provides identifiers for the data items that exist in the system. While the type provides instructions on how to alter the data for the middle-ware application. Thus, DCPS defines which interface to use for interacting with the service [13].

As for DLRL, it locally constructs the data automatically and allows applications to access them. In addition, it provides information about the modified data and updates the local copy. Moreover, it describes the classes of the application as a set of objects along with their methods, data fields, and relations. Hence, some of the data fields provided can be attached to the DCPS entities and used as tags to access those entities. Also, it gives the ability to manipulate objects and manage them in a cache of objects. [13].

B. Security and privacy issues

Because DDS is an application layer, it is vulnerable to multiple threats. That includes unauthorized access to data and devices, data manipulation, and disruption of network connectivity [14]. In particular, possible attacks on DDS protocol can include the misuse of some quality of service (QoS) policies, the misuse of anonymous publish–subscribe functionalities, and the misuse of the LOCATORLIST environment variable in the participant discovery phase. As a side note here, QoS policies are a kind of contract between the publisher and the subscriber that assists aplications in constraining communications without going into detail with the network architecture [15].

First, many QoS policies can be misused in order to attack the DDS protocol including: OWNERSHIP KIND and OWNERSHIP STRENGTH. OWNERSHIP KIND determines how data is distributed. While OWNERSHIP STRENGTH is used to determine which publisher can publish data for a certain topic when the data ownership EXCLUSIVE is used. [15].

In an OWNERSHIP KIND policy, the publisher and the subscriber must have the same data value for the topic in order for communication to occur. Whereas, the data ownership could be either SHARED or EXCLUSIVE [14]. Additionally, a SHARED value indicates that all entities may publish and subscribe to the data. As for an EXCLUSIVE value, only one entity can publish data at a time and only subscribers that have

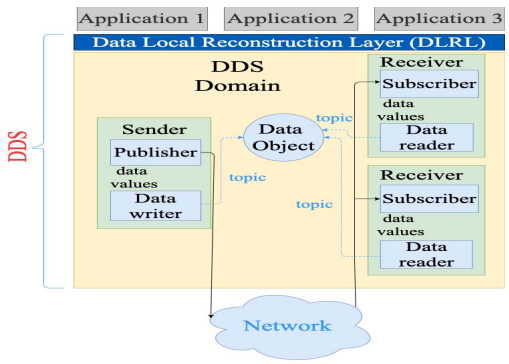

Fig. 2. Conceptual model of DDS. [12].

the same value can subscribe to them. If the ownership value of a publisher is changed, the data will no longer be received by the subscriber. Thus, an attacker may reroute a publisher and takes its place to inject false data [14].

In an OWNERSHIP STRENGTH policy, only one publisher is allowed to publish data with EXCLUSIVE data ownership. Consequently, an attacker may hijack the data by sending it first and preventing the publisher from sending data. Thus, the attacker must know the data type, the QoS policy that is being used, and the current OWNERSHIP STRENGTH value in order to set a higher value. Table 1 shows the discussed policies and their properties [14].

Second, the misuse of anonymous publish–subscribe functionalities takes advantage of the fact that the publish–subscribe mechanism is the center of the DDS protocol. That is, an attacker would try to subscribe to data in the system and then republish or alter the data. Therefore, the attacker can make multiple copies of the data, change the QoS policy of existing data, and alter the position of each copy [16].

Third, the environment variable LOCATORLIST can be misused by directing a new DDS entity to an entirely different DDS domain. That is, the LOCATORLIST variable is used to specify the IP address of the new entity in order to initiate the DDS simple participant discovery protocol (SPDP) [14]. Consequently, SPDP is a discovery protocol that the LOCATORLIST uses to locate and sign any new entities in the network by using the IP address and the port number [15].

C. Solutions

Since the discovery of DDS exploits, many enhancements have been developed to improve its security. For example, the OpenDDS system was designed in order to manage the distributed devices. Furthermore, DDS security standard 1.0 provides encryption, access control, and other security features [17].

An open DDS system adds two new features to the DDS system: secure transport (ST) and discovery service. Additionally, ST divides information based on security classifications. Where the discovery service authorizes and sets up the ST information flow between publishers and subscribers [17].

TABLE I
QoS POLICIES AND THEIR EXPLOITS

Properties	OWNERSHIP_KIND	OWNERSHIP_KIND
Description	Decides which entities can publish and subscribe to data	Decides which publisher will publish the data when EXCLUSIVE is used
Types	SHARED EXCLUSIVE	DDS_Long
Exploits	• Deny subscriber from receiving data • Masquerade as a publisher to send false data	• Deny publisher from publishing data • Send false data

DDS security standard 1.0 is divided into two parts: the security model (SM) and the service plugin interface (SPI) extensions. In addition, SM is a structure that identifies the principals of security, the data that needs to be secured, and the procedures on the restricted data [18]. As for SPI, it is a collection of plugins that are added to the SM model to provide security features, such as authentication, encryption, access control, event logging, and data tagging [18].

Connext DDS Secure™ is an enhanced version of the DDS security standard 1.0 that adds more security features to the system. For instance, it has added public key infrastructure (PKI), certificate authority (CA), digital signature algorithm (DSA) and advanced encryption standard (AES) [15] [19].

IV. MESSAGE QUEUE TELEMETRY TRANSPORT PROTOCOL

MQTT is one of the most popular application protocols. It uses TCP for transport, and it works over IP [20]. Compared to other lightweight protocols, it has more efficient power consumption, especially for constrained resource devices, low bandwidth usage, and smaller header size [21].

A. MQTT Architecture

MQTT is a widely used open standard that works as a pipe for binary data transmission. It based on a publish–subscribe communication pattern for use in resource-constrained devices, such as those with limited memory capabilities and low processing power [20]. The MQTT protocol is widely used for the IoT because it is lightweight and has low bandwidth and cost requirements, even though it has some vulnerabilities such as eavesdropping, modifying accessed data, DoS, re-routing data, etc. [20]. The MQTT protocol transfers messages using a central device (broker) to process the communication between devices as shown in Fig. 3. Each device may subscribe to more than one topic, which is the address for each published message working as a virtual channel. Furthermore, the subscriber will receive every message that was published to these topics. Also, the clients can publish messages to these topics while receiving messages from the same topic or a different one, thus the client can be a publisher and subscriber at the same time [22].

B. Security and privacy issues

For security reasons, the MQTT protocol is authenticated by username and password. In addition, several brokers add various mechanisms along with it. For this reason, the MQTT

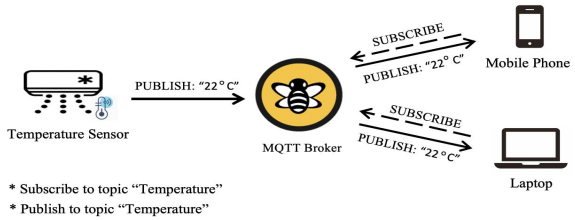

Fig. 3. Example of IoT MQTT broker [23].

security depends on the selection of the broker. Note that, secure sockets layer and transport layer security (SSL/TLS) are mostly used by brokers. However, they have a significant degrading effect on performance for IoT devices [22].

One of the well-known attacks on MQTT is DoS either by flooding the broker with false control or data packets. Therefore, the way to recover from that failure, with the consequences to the broker failure in the IoT applications has raised the need for a competent intrusion detection system (IDS) for the MQTT-based applications [24].

Unicode handling of topic strings can also be a security issue for DoS. Where the developers can choose to close a connection if the validation fails for the rejected UTF-8 code points or not, it is up to them. Hence, an attack can happen if the broker does not implement a periodic check for the rejected UTF-8 code points. However, the client does (or the opposite). Here, a malicious client will benefit from the inconsistency to disconnect other clients by sending wrongly encoded strings [25].

The publish–subscribe scenario with normal clients is shown in (Fig. 4 (a)). Nevertheless, if the malicious client wants to publish a message with invalid UTF-8 data. While the broker follows the standard and implements the check, then this is the best case and there is no issue (Fig. 4 (b)). If the client and the broker do not implement the check, again there is no issue (Fig. 4 (c)). If only the client implements the check, then a disconnection from the network may happen due to an invalid character being received (Fig. 4 (d)). Thus, the clients may be offline for a while due to the re-sending of malicious messages, and the continued disconnection [25].

C. Solutions

A will known countermeasure is the session key generation and distribution. Unfortunately, this reduces the performance of the MQTT protocol. Furthermore, throttling, it prevents the attacker from flooding the broker with false messages. Still, the throttling is inefficient for a large-scale DoS attack in IoT applications. Because of the low configuration and dynamic network features of IoT devices, the traditional detection and prevention methods will not be effective in all IoT network conditions [23].

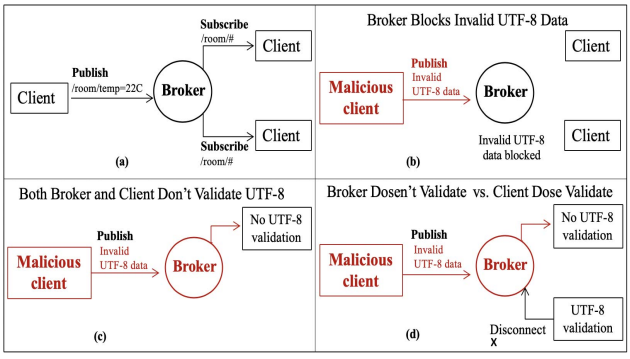

Fig. 4. DoS attacks, Unicode Handling [25].

A lightweight IDS for MQTT-based IoT applications are classified into two, anomaly and signature-based IDS. Signature-based IDS works as a comparison between the previously recognized attack patterns and the behavior of the system at a given moment. Snort is one of the signature-based IDS methods. It was designed for networks with low power consumption. It uses a Boyer-Moore string matching algorithm for the comparison, which works effectively if there is a match set containing unique patterns. Moreover, the database of the attack pattern should be updated continually, otherwise it will not be efficient. In order to detect any malicious activity during communication time, the authors in [23] proposed Secure-MQTT, which is a new lightweight detection scheme using fuzzy logic-based. They proved that it is lighter weight than MQTT and has better performance because the fuzzy rule interpolation mechanism generates rules dynamically.

V. Long Range Wide-Area Network Protocol

The LoRaWAN technique is suggested to maximize LP-WANs for power consumption, quality, efficiency, and scope. The following subsection explains the LoRaWAN architecture, security and privacy issues, and enhanced solutions.

A. LoraWan Architecture

As shown in Fig. 5, the LoRaWAN architecture consists of gateways, end nodes, application server, and network server. The end nodes forward data to one or more gateways via the LoRa physical layer. Then, every gateway transmits data through end nodes to the network server employing every backhaul (wired, Wi-Fi, satellite, or Ethernet). The network server is the intelligent unit that controls the network, conducts security checks, filters redundant packets, etc. [26].

B. Security and privacy issues

The LoRaWAN network is an emerging research field, with benefits and an expanded reach of improvements. The LoRaWAN network, despite being well-designed, faces many security and privacy issues. LoRaWAN uses standard security characteristic. But suffers from the following weaknesses [27].

- The joining procedure causes vulnerabilities, which result in replay attacks being exploited.
- The protocol cannot provide end-to-end security. As the session key for each device and its application server

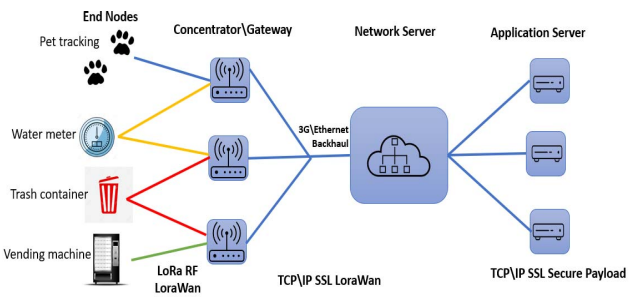

Fig. 5. LoraWan Architecture [26].

is configured via core network support. Put differently, the LoRaWAN network processor can easily distinguish between the two parties.
- The application and network session keys are based on a shared long-term key and cannot provide precise forward confidentiality. In regard to the fact that each system can be broken and compromised, the long-term key can also be displayed, producing past session keys, and retrieving their encrypted data.

The authors in [28] presented five weaknesses and defined specific attacks that might abuse them. These attacks were the replay attack for ABP-activated nodes, the bit-flipping attack, eavesdropping, LoRa class B attacks, and ACK spoofing. In this paper, we will mention in detail the first attack. End devices must be activated so they can connect to the network server. In a LoRaWAN network, there are two ways to activate end devices: Activation By Personalisation (ABP), and over-the-air activation (OTAA). ABP-activated end devices are pre-programmed with fixed keys on each device. Also, the v1.0.2 protocol specifier states, "After an exchanged or restarted JoinReqJoinAccept message personalizes each end user, the end device frame counters and the network service frame counters are reset to 0."

The ABP activated end device returns the frame counter value from 0 with the same keys. In this scenario, during the last session, the attacker will repeat messages with higher counter values. With the exception of reset, counter overflow is a straightforward way to restart the counter. When the counter at its highest value, the value will be reset to 0. The intruder may even replay previous messages with the counter values of the last session and the same session keys to interrupt contact between the server and the end device. This refers to both ABP and OTAA [28].

In comparison, the intrusion on an ABP-activated end device would take less time to both overrun and reset if the attacker were willing to restore the end devices. LoRaWAN conducts AES channel confidentiality in counter mode. Besides, the value of the packet counter should be viewed as an input to define the counter as a nonce. When the counter value is reset while the key is still in place, the block cipher recreates exactly the same key content. Hence, the reuse of the keystream is the normal case [28].

C. Solutions

The authors in paper [27] proposed an improved LoRaWAN security protocol that fixes the security vulnerabilities of the standard. Not only does it recommend the DevNonce creation method to prevent replay attacks, it also helps a computer and its application server to effectively achieve end-to-end security via the key exchange depending on the elliptic-curve diffie–hellman (ECDH) scheme. The proposed procedure in fact offers two options: the security-enhanced option (SEO) and default option (DO). Therefore, it prevents breaches of the end-to-end security of both the system and its application server using a compromised network server. The suggested protocol presupposes that:

- AppKey is a key exchanged between the system and its network server over the long term.
- A protected connection between the network and the application servers is pre-established.
- Every device has and trusts the ECDSA public key PUAPP for its application server, which is used to check the digital signature of the application server.
- For SEO, each device should has its own ECDSA public key pair, and the application server should trust its public key PUDEV.

The authors in [28] suggest measures to prevent a replay attack from happening, which are:

- The use of configuration activation should be reduced, and new keys should be installed regularly if used.
- End system must be physically secured to prevent unauthorized parties from triggering a reset of the network.
- The end system needs to re-key every time the counter exceeds its highest value to avoid a replay attack.

The eavesdropping attack targets the cipher block in counter mode, which is not safe if the counter value is permitted to repeat it. This issue can be solved in a number of ways:

- Substitute the stand value with a nonce (an amount utilized just once) generated from a cryptographic protected pseudo-random number dynamo or a real arbitrary number dynamo on the sensor network.
- Re-key on reset, as well as on any counter overload. When this attack involves the collection of multiple messages with identical session keys and counter values, regularly replacing session keys will prevent the intruder from getting sufficient messages.

VI. Discussion

In this study, we presented a review of some of the IoT protocols most often used in the application and the network layer. These are: DDS, MQTT, and LoRaWAN. We have focused in this paper on issues and vulnerabilities related to privacy in these protocols. From our point of view, we see that they are hot topics for research and open for improvement to overcome security issues. Table II below, shows a comparison between the three protocols in terms of protocol type, transport, architecture, security, properties, and security issues.

VII. Conclusion

In this study, we have provided an overview and discussion some of the IoT protocols that are widely used in the application and the network layer. Hence, we reviewed DDS and MQTT protocols in the application layer, and LoRaWAN protocol in the network layer. Also, we discussed their architecture, some of the main security issues, and some proposed solutions.

DDS showed security breaches in the functionalities of the publish-subscribe model, QoS policies, and in altering the environment variables. Thus, the protocol security was improved through multiple approaches including open-DDS system, DDS security standard 1.0, and Connext DDS secure™. As for MQTT, we found that this protocol was

vulnerable to DoS attacks, and like the DDS protocol, MQTT is also vulnerable to environmental variables altering. However, the session key and throttling could be a solution for these vulnerable. Also, a new enhanced version is introduced called the Secure-MQTT. LoRaWAN, on the other hand, it faces some issues with the exposure of keys, the inability to provide secure end-to-end connection. Hence, it is exposed to replay attacks on ABP-activated nodes, eavesdropping, bit-flipping attacks, ACK spoofing, and LoRa class B attacks. Therefore, some of the solutions that were introduced include using AppKey for key exchange, pre-establishing a secure connection, reducing the use of configuration activation, and replacing keys whenever the counter exceeds its highest value.

TABLE II
COMPARISON BETWEEN DDS, MQTT,AND LORAWAN PROTOCOLS

	DDS	MQTT	LoRaWan
Protocol	Application Layer Protocol	Application Layer Protocol	Network Layer Protocol
Transport	UDP	TCP	TCP , UDP
Architecture	Publish/subscribe Data centric	Publish/Subscribe	End to End Secure Payload
Security	AES-GCM for encryption AES-GMAC for message authentication	TLS/SSL	AES 128 bit
Properties	• Highly scalable • Applicable to any type of communication such as P2P, device-cloud, etc. • Efficient in resources consumption	• Lightweight messaging protocol • For resource-constrained devices • low bandwidth and cost requirements	• Battery lifetime reduction • High network capacity • Different device classes
Security issues	• Unauthorized access to data and devices • Data manipulation • Network disruption • Eavesdropping	• DoS attack –Flooding the broker with false control • Malicious code – Unicode handling of topic strings • Eavesdropping	•Joining procedure •End-to-end security •Replay attack for ABP-activated nodes •Bit-flipping attack •Eavesdropping •LoRa class B attacks •ACK spoofing.

REFERENCES

[1] Noah Apthorpe, Dillon Reisman, and Nick Feamster. A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic. May 2017.

[2] Qusay Hassan. Introduction to the internet of things. In *Internet of Things A to Z: Technologies and Applications, IEEE*, pages 1–50. 2018.

[3] Omar Alhazmi and Khalid Aloufi. Fog based internet of things a security scheme. In *2nd International Conference on Computer Applications Information Security (ICCAIS)*. May 2019.

[4] Muhammad Burhan, Rana Asif Rehman, Bilal Khan, and Byung-Seo Kim. Iot elements, layered architectures and security issues: A comprehensive survey. May 2019.

[5] Bedir Tekinerdogan Omer Koksal. Obstacles in data distribution service middleware: a systematic review. pages 191–210. 2017.

[6] Meena Singh, Rajan MA, Shivraj VL, and Balamuralidhar P. Secure mqtt for internet of things (iot). In *Fifth International Conference on Communication Systems and Network Technologies*. August 2018.

[7] ADNAN M. ABU-MAHFOUZ JACO MORNÉ MARAIS and GERHARD P. HANCKE. A survey on the viability of confirmed traffic in a lorawan. pages 9296–9311. 2020.

[8] Ruffin White, Gianluca Caiazza, Chenxu Jiang, Xinyue Ou, Zhiyue Yang, Agostino Cortesi, and Henrik Christensen. Network reconnaissance and vulnerability excavation of secure dds systems. pages 57–66. June 2019.

[9] Syed Firdous, Zubair Baig, Craig Valli, and Ahmed Ibrahim. Modelling and evaluation of malicious attacks against the iot mqtt protocol. In *IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. June 2017.

[10] Robert Miller. Lora security - building a secure lora solution. 2016.

[11] Christian Lübben Georg Aures. Dds vs. mqtt vs. vsl for iot. 2019.

[12] Turgay Celik Omer Koksal, Bedir Tekinerdogan. Data distribution service-based architecture design for the internet of things systems. pages 269–285. 2017.

[13] Object Management Group (OMG). About the dds security specification version 1.1. https://www.omg.org/spec/DDS-SECURITY/About-DDS-SECURITY/.

[14] Michael Michaud, Thomas Dean, and Sylvain Leblanc. Attacking omg data distribution service (dds) based real-time mission critical distributed systems. In *13th International Conference on Malicious and Unwanted Software (MALWARE). IEEE*, pages 68–77. October 2018.

[15] Michael Michaud, Sylvain Leblanc, and Thomas Dean. Malicious use of omg data distribution service (dds) in real-time mission critical distributed systems. In *IEEE communications surveys tutorials*. 2017.

[16] Imran Makhdoom, Mehran Abolhasan, Justin Lipman, Ren Ping Liu, and Wei Ni. Anatomy of threats to the internet of things. In *IEEE Communications Surveys Tutorials*, pages 1636–1675. October 2018.

[17] Object Computing Incorporated (OCI). Opendds. https://opendds.org/.

[18] Object Management Group (OMG). Dds security version 1.1. July 2018.

[19] William Stallings, Lawrie Brown, Michael Bauer, and Arup Bhattacharjee. In *Computer security: principles and practice.* May 2017.

[20] Muneer Bani Yassein, Mohammed Shatnawi, Shadi Aljwarneh, and Razan Al-Hatmi. Internet of things: Survey and open issues of mqtt protocol. In *International Conference on Engineering MIS (ICEMIS)*. May 2017.

[21] Özlem YERLİKAYA and Gökhan DALKILIÇ. Authentication and authorization mechanism on message queue telemetry transport protocol. In *International Conference On Telecommunication, Power Analysis And Computing Techniques*. Sept. 2018.

[22] Dipa Soni and Ashwin Makwana. A survey on mqtt: a protocol of internet of things (iot). In *International Conference on Engineering MIS (ICEMIS)*. 2017.

[23] Haripriya AP and Kulothungan K. Secure-mqtt: an efficient fuzzy logic-based approach to detect dos attack in mqtt protocol for internet of things. pages 1–15. 2019.

[24] Hasan AliKhattak, Munam Ali Shah, Sangeen Khan, Ihsan Ali, and Muhammad Imran. Perception layer security in internet of things. volume 100, pages 144–164. November 2019.

[25] Federico Maggi, Rainer Vosseler, and Davide Quarta. The fragility of industrial iot's data backbone. 2018.

[26] Sarra Naoui, Mohamed Elhdhili, and Leila Saidane. Enhancing the security of the iot lorawan architecture. 2016.

[27] Ilsun You, Soonhyun Kwon, Gaurav Choudhary, Vishal Sharma, and Jung Seo. An enhanced lorawan security protocol for privacy preservation in iot with a case study on a smart factory-enabled parking system. Sensors 2018.

[28] Xueying Yang, Evgenios Karampatzakis, Christian Doerr, and Fernando Kuipers. Security vulnerabilities in lorawan. pages 129–140. 2018.