

JURNAL INFOTEL Informatics - Telecommunication - Electronics Website: http://ejournal.st3telkom.ac.id/index.php/infotel ISSN: 2085-3688; e-ISSN: 2460-0997



A study of secure communication scheme in MQTT: TLS vs AES cryptography

Favian Dewanta^{1,*}, Baiq Yuniar Yustiarini¹, Bangun Indrakusumo Radityo Harsritanto²
¹School of Electrical Engineering, Telkom University
²Architecture Department, Universitas Diponegoro
¹Jl. Telekomunikasi, No. 1, Bandung 40257, Indonesia
²Jl. Prof. Soedarto, Semarang 50275, Indonesia
*Corresponding email: favian@telkomuniversity.ac.id

Received 12 July 2022, Revised 20 September 2022, Accepted 22 September 2022

Abstract — The Internet of Things (IoT) technology help devices to send command and request, exchange messages, and also communicate with an entire level of devices and infrastructures. Several IoT-based systems may communicate at a certain level of latency depending on the urgency and purpose of the communications. Therefore, several protocols with respect to their speed and reliability levels can be chosen for achieving the desired quality of services. As for lightweight and fast communication speed for IoT devices, the MQTT protocol is the most commonly known and recommended in the system. However, the MQTT protocol is not equipped with the appropriate security mechanism. As a consequence, the MQTT messages are easily eavesdropped on and modified by the attackers during communication sessions among devices through several levels of network domains including local, internet, and internal cloud-based network. Considering the well-established security approach and commonly strong cipher system, this research studies the use of the AES cryptographybased communication scheme against the TLS-based communication scheme, which can be used to create end-to-end secure communication scheme possesses the highest cost in terms of communication delay and network cost among all schemes in the experiment. Eventually, the AES-based MQTT communication scheme is more appropriate for IoT environments because of its communication delay and network cost, which are considered equal to the plaintext-based MQTT communications.

Keywords - lightweight cryptography, MQTT, secure communication

Copyright ©2022 JURNAL INFOTEL All rights reserved.

I. INTRODUCTION

Internet of Things (IoT) technology requires low computational devices, either CPUs, memories, or storage. Those requirements are caused by having IoT devices to conduct specific tasks in an energylimited area, such as sensing humidity, temperature, soil moisture in the farm, forest, river, mountain, and so on. Therefore, the components for implementing such mentioned systems are limited by the computational and energy resources and physical dimension with respect to higher functionality and longer device lifetime [1].

Regarding the issue of limited computational resources, Mishra and Kertesz [2] mention several popular communication protocols, which are CoAP, MQTT, HTTP, AMQP, and so on. However, they prefer the MQTT due to its simplicity and easiness of installing that protocol. In addition, the MQTT also provides several QoS for maintaining high-quality communication with respect to packet size and latency.

However, the MQTT is not equipped with a security feature in which the MQTT messages can easily be eavesdropped on and stolen by attackers. Moreover, there are several IoT applications that demand high privacy and confidentiality, such as telemedicine, Smart Home, Smart Mosque, and Industrial Plant. Thus, security issues in the MQTT protocol become a challenge for establishing trusted IoT applications [3].

With respect to the security issues in the MQTT protocol, several researchers and practitioners propose the TLS for providing confidentiality and authenticity [4], [5]. However, even though the TLS is proven to deliver services in a secure and trusted

	127 28.364522	127.0.0.1	127.0.0.1	MQTT	61 Publish Message [temp/12]
	128 28.364537	127.0.0.1	127.0.0.1	TCP	44 1883 → 59752 [ACK] Seq=5 Ack=32 Win=2619648 Len=0
1	129 28.364551	127.0.0.1	127.0.0.1	TCP	45 59750 → 59751 [PSH, ACK] Seq=3 Ack=1 Win=2619648 Len=1
	130 28.364561	127.0.0.1	127.0.0.1	TCP	44 59751 → 59750 [ACK] Seq=1 Ack=4 Win=2619648 Len=0
<					
> Null/L	oopback				
> Interr	net Protocol Version 4.	Src: 127.0.0.1, D	st: 127.0.0.1		
> Trans	mission Control Protocol	. Src Port: 59752	. Dst Port: 1883, Sea:	15. Ack: 5.	Len: 17
✓ MO Tel	lemetry Transport Protoc	ol, Publish Messa	ge		
> Hea	der Flags: 0x30, Messag	e Type: Publish M	essage, OoS Level: At r	most once del:	ivery (Fire and Forget)
Mse	Len: 15				
Tor	bic Length: 7				
Tor	ic: temp/12				
Mee	cage: 31322e383643				
	Suge: 515220505045				
0000 02	00 00 00 45 00 00 39	43 6f 40 00 80 06	00 00 ····E··9 Coll-		
0010 7f	00 00 01 7f 00 00 01	e9 68 07 5b a6 06	97 9f		
0020 4b	a4 51 bd 50 18 27 f9	9d 41 00 00 30 Of	00 07 K.Q.P.'. A.	a	
0030 74	65 6d 70 2f 31 32 31	32 2e 38 36 43	temp/121 2.860	C	

Fig. 1. The message of MQTT protocol from a publisher to the Mosquitto broker located on local host.

manner, such as internet banking and other financial technology platforms, the TLS is considered a heavy and high latency platform for securing IoT devices with low computational resources.

Several other researchers then propose the use of password-based authentication key exchange (PAKE) as an additional security component of the MQTT [6]–[8]. However, the PAKE methods are considered a partial solution because the data are stored in the MQTT broker in a plaintext condition. Moreover, the use of asymmetric cryptography, point multiplication, and the power of integer techniques are still problematic for IoT platforms due to the high computational and communication cost.

Another approach based on the covert channel mechanism is proposed by Velinov *et al.* [9]. It is proven to be effective to hide secret information by means of direct and indirect covert channels utilizing a modification of topic name, packet identifier, and application message. However, the covert channel approach leads to bandwidth degradation with respect to covert channel placement. Therefore, it can cause high latency and eventually disturb the service of real-time IoT applications. Therefore, the use of covert channel approaches is not favorable in the common situation, unless IoT applications allow slow response and high latency communication.

This paper discusses the issues of a secure communication channel in the MQTT for providing trusted IoT applications. Specifically, this paper does not discuss authentication approaches, but it limits the discussion to the use of AES [10] and TLS scheme [11] because this work is the continuation of our previous work, which discusses authentication approaches between the PAKE model and XOR and hash model [12]. Both the AES and TLS are selected in this research due to the popularity and feasibility of those two approaches for securing communication in any high-layer network protocols, including the MQTT communication protocols. Then, the contributions of our work are as listed below.

1) We explain two MQTT broker platforms

(Mosquito and Hivemq) and their weaknesses in section II.

- 2) We describe our research methodology and experimental setup in section II.
- 3) Lastly, our experimental results and analyses are delivered in section III.

II. RESEARCH METHOD

There are three discussions in this section, which are the MQTT messaging issues, the comparison of AES and TLS approaches in MQTT communication, and the experimental setup for measuring the performance of both approaches.

A. The MQTT Messaging Issues

The MQTT protocol [2], [9], [13] is one of the IoT messaging platforms based on publish and subscribe operations. The protocol consists of several entities, which are (1) the publisher as a source of the message, (2) the subscriber as a sink of the message, and (3) the broker as a hub among the publisher and the subscriber. In order to set a communication channel between the publisher and the subscriber, those components should publish and subscribe to the same topics in the broker.

There are several scenarios for implementing MQTTbased communication in our IoT projects. One may install the MQTT publisher in the IoT devices acting as sensors by means of several programming languages. In another case, IoT devices acting as actuators may be set as the MQTT subscriber. As for the broker, sometimes it can be installed on cloud or/and edge computing servers. The locations and setup scenarios of those entities, *e.g.*, publisher, subscriber, broker, depending on the design of the IoT systems and sometimes the stakeholders of the IoT projects.

Regarding the MQTT broker platforms, the IoT project developers can utilize several available platforms rather than implementing the broker from the scratch by means of several programming languages. There are several freely available MQTT brokers, which are Mosquitto, HiveMQ, RabbitMQ, EMQ X, and so on. By using those MQTT broker platforms, IoT developers merely need to adjust the topic name and the address

(IP and ports) of those brokers in their IoT devices and cloud servers acting as the publishers and subscribers.

As discussed previously, the MQTT protocol is not equipped with security features. In the basic operation of the MQTT protocol as hosted by the Mosquitto broker, the MQTT messages are sent in the plaintext form as given in Fig. 1 in which a publisher sends an MQTT message containing temperature information '12.86C' with the topic 'temp/12' to the broker. As a consequence, those MQTT communication channels are vulnerable to any attack scenarios, including information theft, man-in-the-middle attack, reply attack, and *so on*.

As for the HiveMQ broker and/or any other enterprise-ready MQTT broker platforms, they are frequently equipped with certain message encoding based on the topic name. As a consequence, their MQTT messages seem to be encrypted as given in Fig. 2. However, if attackers can adjust their subscribers to the same topic name as the targeted communication channel between publishers and subscribers, the attackers can eventually get the plaintext form of the MQTT messages. In Fig. 3, we use the websockets client provided by the HiveMQ platforms to capture our previously encoded MQTT messages in Fig. 2 by means of adjusting the topic name. Thus, this kind of MQTT messaging mode in the HiveMQ broker and any other similar platforms is only effective for protecting the system from random outsider attacks, but not from insider attacks knowing the topic of the MQTT messaging systems.

Considering these issues of MQTT messaging, this paper proposes to use a symmetric cryptography method to protect the MQTT messages end-to-end from the publishers to the subscribers. Eventually, the study of employing AES cryptography here can be viewed as an alternative solution for providing secure communication in the MQTT protocol with respect to the TLS approach in the MQTT communication.

B. The Comparison of AES and TLS Approaches

One may wonder whether the AES and TLS approaches are comparable in this case because the TLS is a complete security system including authentication, key agreement, and encryption-decryption schemes based on message handshakes between two entities [14], [15]. The AES encryption-decryption scheme is certainly also part of the TLS scheme. However, we can argue that the AES and TLS schemes are still comparable in a way that the MQTT communication may select those two schemes based on the specification of the IoT devices. Specifically, the IoT devices with low computational resources may utilize the AES scheme rather than the TLS with the assumption that both MQTT publisher and subscriber already share the same key and encryption-decryption mode, i.e., ECB, CBC, CFB, etc.

In order to possess identical keys and encryptiondecryption modes, there are two approaches that can be taken, which are fixed pre-shared key (PSK) that is similar to WEP in WiFi security system [16], [17] and authenticated key exchange (AKE) approaches [18], [19]. The fixed PSK approach is the simplest and easiest way, but it is more vulnerable with respect to the AKE approaches due to the use of fixed key and its consequences against several known attacks, such as brute force attacks, replay attacks, *etc*.

Regarding the TLS scheme in the MQTT communication, the MQTT agents (publishers and subscribers) can activate the TLS feature by using a public key certificate (PKC) based on X.509 digital certificate. In some MQTT broker platforms, *e.g.*, in RabbitMQ, HiveMQ, *etc.*, the SSL/TLS feature has been bundled with the MQTT broker and agents. Thus, the TLS can be executed seamlessly by using some codes provided in Python, NodeJS, and so on. However, several MQTT broker platforms do not provide the SSL/TLS feature. In this case, users need to install the OpenSSL application including the digital certificate based on X.509.

In this research, our experiment will compare the performance of MQTT communications utilizing the AES encryption-decryption scheme against the TLS scheme. Considering the TLS feature as mentioned in the previous paragraph, we decide to utilize HiveMQ broker for this experiment. As for the AES cryptography scheme, this experiment uses the AES with the key sizes 128, 192, and 256 bits.

C. MQTT Experimental Setup

The experiment is done by using the network topology as given in Fig. 4. There is an MQTT broker/server connecting an MQTT publisher and an MQTT subscriber through the internet. As for the MQTT server, we use HiveMQ broker in the TLS and non-TLS modes. As for the MQTT publisher and subscriber, we use the Paho-MQTT library based on Python programming language to implement those MQTT agents.

For increasing clarity of the experiment, the illustration of the MQTT message is given in Fig. 5. The MQTT message is encapsulated by headers in TCP, IP, and Ethernet layers respectively. Therefore, the total frame of the MQTT message will be

$$L_f = 54 + x \tag{1}$$

in which variable x represents the length of the MQTT message (bytes).

As for the MQTT message, it is located in the OSI layer 7 and it has a specific format as given in Fig. 5. The MQTT format consists of header flags (1 byte), message length (1-4 bytes), variable length header (0-Y bytes), and message/content (0-Z bytes). The variable

	3079 78.128725	192.168.1.6	35.157.252.226	TCP	62 60411 → 8000 [PSH, ACK] Seq=251 Ack=9 Win=512 Len=8
	3080 78.323118	35.157.252.226	192.168.1.6	TCP	54 8000 → 60411 [ACK] Seq=9 Ack=251 Win=110 Len=0
	3081 78.323438	35.157.252.226	192.168.1.6	TCP	54 8000 → 60411 [ACK] Seq=9 Ack=259 Win=110 Len=0
	3082 78.323438	35.157.252.226	192.168.1.6	TCP	58 8000 → 60411 [PSH, ACK] Seq=9 Ack=259 Win=110 Len=4
	3083 78.323438	35.157.252.226	192.168.1.6	TCP	58 8000 → 60411 [PSH, ACK] Seq=13 Ack=259 Win=110 Len=4
L	3084 78.323485	192.168.1.6	35.157.252.226	TCP	54 60411 → 8000 [ACK] Seq=259 Ack=17 Win=512 Len=0
<					
> Tr Y Da	ransmission Control Protocol ata (45 bytes) Data: 82a778eec94d48cbc942 [Length: 45]	, Src Port: 60411, D: 0c8bba390c81b9241bc1k	st Port: 8000, Seq: 5 b83a49dcfa391d83b9774	5, Ack: 5, 9dce77a4ea	Len: 45 d
0000	00 67 62 64 d2 18 34 f3	9a 00 32 a0 08 00 45	00 ·gbd · · 4 · · · 2 · · · 1		
0010	00 55 31 ee 40 00 80 06 e	26 86 c0 a8 01 06 23	9d .01.@	ŧ.	
0020	fc e2 eb fb 1f 40 21 b3 0	c4 a9 05 ba 46 ta 50	18 ·····@!· ····F·		
0030	02 00 e3 61 00 00 82 a7	78 ee c9 40 48 cb c9	42 ···a···· x··MH·	в	
0040	1d 92 b0 77 40 dc o7 75	LU CI UO 38 49 0C TA	77	9	
0050	Af de ec	+e au eo oz 10 90 a4	Out Out	W	
0000	TI WE EE		•		

Fig. 2. The message of MQTT protocol from a publisher to the HiveMQ broker.

P

🚯 HIVE MQ			Websockets Cl	ient Showc
Connection			connected	≫
Publish		~	Subscriptions	~
Topic testtopic/1	QoS Retain	Publish	Add New Topic Subscript	ion
Message			Cos: 2 testtopic/qw123	×
Messages		*		
2022-06-20 06:30:51 Topic: testtopic/gw123 temp:12.76C//hum:70%	Qos: 0			
2022-06-20 06:30:50 Topic: testtopic/qw123 temp:12.76C//hum:70%	Qos: 0			

Fig. 3. The captured messages from the encoded MQTT communication channel by adjusting topic name of the HiveMQ websockets client.



MQTT Subscriber

Fig. 4. The MQTT communication system used in this experiment.

length header and message/content may exist in the MQTT message due to the existence of the MQTT control packet type that is described in the header flags. Further information regarding the MQTT control packet type can be checked in the following articles [2], [20].

For measuring the performance of those MQTT communication schemes (AES vs TLS), we evaluate several parameters as described below.

• Communication delay (D_{comm}) is measured by calculating the time difference between MQTT message transmission and reception. For getting the time difference, we put the timestamp in



Fig. 5. The MQTT message format in the point of view OSI layer.

seconds in the MQTT message as given in Fig. 6 by using the Python time() method. Theoretically, the communication delay can be calculated by accumulating all components of delays in the communication process or network. In this case, a component of delays will be transmission delay (T_d) and queueing delay (Q_d) from the publisher to the subscriber. The transmission delay is the delay to transmit frames through network media and the queueing delay is the processing delay in the network devices.

$$D_{comm} = \sigma T_d + \sigma Q_d \tag{2}$$

• Computational delay is measured by calculating the time difference between the beginning and end of executing certain methods. In this case, we compute the computation delay of executing the AES encryption and decryption method. As for



Fig. 6. The use of timestamp in the MQTT message for measuring communication delay from the publisher to the subscriber.

the plaintext and TLS schemes, the computational delay is equal because the MQTT computational process only handles the plaintext data for both plaintext and TLS schemes and the TLS is handled by the process in the lower layer (network kernel).

• Network cost is the number of packets in the network, in maintaining communication between publishers and subscribers. These packets include the MQTT message, TCP signaling packet, TLS signaling packet, *etc*.

No.	Parameter	Value
1	Number of MQTT message	100
	samples per scenario	
2	AES key size	128, 192, and
		256 bits
3	MQTT payload size in plaintext	20, 30, and 40
		bytes
4	MQTT message inter-departure time	2 and 5 seconds
5	Network bandwidth	Up to 10 Mbps

Table 1. Parameters of Experiment

The experimental parameters are described in Table 1. As for the number of MQTT message samples, we use 100 samples to anticipate the unpredictability of the network traffic due to using public network infrastructure with up to 10 Mbps bandwidth. The AES key size is set to be 128, 192, and 256 bits to the given plaintext payloads which are 20, 30, and 40 bytes. At last, the MQTT message inter-arrival is set to be 2 seconds for anticipating long delays due to the high network traffic and HiveMQ server occupation.

III. RESULT

Before discussing the performance evaluation of both the TLS and AES, we need to show the captured packet by both schemes to demonstrate that both schemes provide confidentiality and secrecy for exchanged MQTT messages. Fig. 7 shows the captured TLS traffic using the Wireshark application. In that figure, the type of the packet above the TLS layer including its content is invisible to the attacker. As for the AES scheme in Fig. 8, the type of the packet, in which the MQTT packet, can be identified clearly by the Wireshark. However, the content of the message is covered by the AES encryption so that the content shown in Fig. 8 seems like random characters. Those two figures eventually prove that the TLS and AES schemes can secure the MQTT message content.

The experimental result for communication delay is shown in Fig. 9. There are three scenarios of payload sizes (20, 30, and 40 bytes) and MQTT communication schemes (Plaintext, AES, and TLS data transmissions). The communication delays vary from 0.31 to 0.392 seconds and there is no certain pattern that differentiates the communication delays for all payload sizes and communication schemes, except for the TLS scheme which shows slightly higher communication delay with respect to the plaintext and AES communication scheme. These communication delays are also supported by the frame size results, with respect to payload size, captured by Wireshark as shown in Fig. 10. It is clear in that figure that the frame size of TLS communication schemes is the highest one among those schemes.

However, we can see that these results generally emphasize that the significant component contributing to communication delays of those 3 payload sizes and communication schemes among MQTT publishers, broker, and subscribers are the queueing delay. The transmission delay generated from the frame size affects the communication delays, but the contribution of the transmission delay is not significant with respect to queueing delay due to using a public network with random characteristics and user behaviors. Therefore, the communication delay in this experiment is mostly driven by the network quality and there is no significant correlation between the payload size and communication delay.

Regarding the computational delay, Fig. 11 shows that AES-256 tends to have the most efficient computational delay with respect to AES-128 and AES-192. However, those delays are not significant due

No.	Time	Source	Destination	Protocol	Length Info
	76 10.829786	113.29.117.24	192.168.1.7	TLSv1.2	204 Server Hello, Change Cipher Spec, Encrypted Handshake Message
	77 10.830382	192.168.1.7	113.29.117.24	TLSv1.2	105 Change Cipher Spec, Encrypted Handshake Message
	78 10.830684	192.168.1.7	113.29.117.24	TLSv1.2	107 Application Data
	79 10.830772	192.168.1.7	113.29.117.24	TLSv1.2	110 Application Data
	80 10.830846	192.168.1.7	113.29.117.24	TLSv1.2	96 Application Data
	81 10.831054	192.168.1.7	113.29.117.24	TLSv1.2	195 Application Data
	82 10.831682	192.168.1.7	113.29.117.24	TLSV1.2	161 Application Data
	83 10.970541	113.29.117.24	192.168.1.7	TLSV1.2	107 Application Data
	04 10.970541 95 10 070541	113.29.117.24	192.100.1.7	TCP	56 443 > 61523 [ACK] Seq=151 ACK=569 Win=64232 Len=0
	86 10 970541	113.29.117.24	192.108.1.7	TCP	56 443 + 61523 [ACK] Seq=151 ACK=509 Win=64232 Len=0
	87 10 970541	113 29 117 24	192.168.1.7	TLSv1 2	109 Application Data
	88 10.970541	113.29.117.24	192.168.1.7	TL Sv1.2	164 Application Data
	89 10.970541	113.29.117.24	192.168.1.7	TL Sv1.2	124 Application Data
	90 10.970724	192.168.1.7	113.29.117.24	TCP	54 61523 → 443 [ACK] Seg=1019 Ack=204 Win=64037 Len=0
	91 10.970823	192.168.1.7	113.29.117.24	тср	54 [TCP Dup ACK 90#1] 61523 → 443 [ACK] Seq=1019 Ack=204 Win=64037 Len=0
> Fra	me 81: 195 bytes	on wire (1560 bits).	195 bytes captured	1560 bit) on interface \Device\NPF {654F777B-4E6F-47B4-BCE9-373F602802A7}, id 0
> Eth	ernet II. Src: C	honggin 91:f1:5f (e8:	6f:38:91:f1:5f), Dst:	zte ad:	:e:8c (44:fb:5a:ad:ce:8c)
> Int	ernet Protocol V	ersion 4. Src: 192.16	8.1.7. Dst: 113.29.11	17.24	(
> Tra	nsmission Contro	1 Protocol, Src Port:	61523, Dst Port: 443	3, Seq: 7	71, Ack: 151, Len: 141
> Tra	nsport Layer Sec	urity			
0000	44 fb 5a ad ce 8	3c e8 6f 38 91 f1 5f	08 00 45 00 D.Z	·o 8·· ··	E.
0010	00 b5 1a 64 40 0	00 80 06 37 fa c0 a8	01 07 71 1d ···· d@·	7	q.
0020	75 18 f0 53 01 b	ob 58 65 ba f4 dd b3	90 b0 50 18 u··S··	Xe	p.
0030	fa 5a 17 a0 00 0	00 17 03 03 00 88 5c	83 56 55 3a ·Z····		U:
0040	10 04 95 CI 49 6	52 1e e7 4e 3d 2b ae	1 30 a2 311.	N-+	
0060	16 0f aa cc 0b 6	5f 1b 51 4a 2a 4d 2e	a7 0a b3 b4	•0 J*M. ••	
0070	b9 16 4e 10 0d d	d1 94 d2 40 ad 17 4b	e9 9b 87 71 ···N····	•• @••к••	·q
0080	ee 14 1d f6 51 9	93 b1 95 17 86 33 97	6a f0 3f 04Q.	·· ··3·j·	?.
0090	6b 49 2c ed 1b b	04 cf bc 5b 20 e9 28	04 ec 6d e6 kI,	[.(m•
00a0	/1 9c 04 a4 1b 5	b dd 49 b5 c2 c8 f0	8a 11 56 /9 g	.1	VV
aaba	6h h2 17 ap ad 6	00 c4 ch 00 E6 67 E1	EA OF OF AN LAND	·····	
00b0	6b b3 17 aa ad 9 f5 fa cb	98 c4 cb 08 56 67 51	54 8f 85 4a k	·· ·VgQT·	ć.
00b0 00c0	6b b3 17 aa ad 9 f5 fa cb	98 c4 cb 08 56 67 51	54 8f 85 4a k	••••VgQT•	·j
00b0 00c0	6b b3 17 aa ad 9 f5 fa cb	98 c4 cb 08 56 67 51 Fi	54 8f 85 4a k	····vgQT·	-j
00b0 00c0	6b b3 17 aa ad 9 f5 fa cb	98 c4 cb 08 56 67 51 Fi	54 8f 85 4a k g. 7. The MQTT	traffic	protected by the TLS scheme.
00b0 00c0 No.	6b b3 17 aa ad 9 f5 fa cb Time	98 c4 cb 08 56 67 51 Fij	54 8f 85 4a k g. 7. The MQTT	••••vgQT•• • traffic Protoc	ri protected by the TLS scheme.
00b0 00c0 No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534	88 c4 cb 08 56 67 51 Fig Source 192.168.1.9	54 8f 85 4a k g. 7. The MQTT Destination 239.255.255.250	traffic	of Length Info
00b0 00c0 No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147	88 c4 cb 08 56 67 51 Fi Source 192.168.1.9 192.168.1.7	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245	••••VgQT•• • traffic Protoc SSDP MQTT	protected by the TLS scheme.
00b0 00c0 No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011	88 c4 cb 08 56 67 51 Fig Source 192.168.1.9 192.268.1.7 3.122.42.245	54 8f 85 4a k g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7	 VgQT traffic Protoc SSDP MQTT MQTT 	-j protected by the TLS scheme. ol Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id=2) [test/1] 107 Publish Message (id=101) [test/1]
00b0 00c0 No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011	Fig. 2014 Colored Colo	54 8f 85 4a k g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 192.168.1.7	traffic Protoc SSDP MQTT MQTT MQTT	<pre>-j protected by the TLS scheme. ol Lengh Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id=2) [test/1] 107 Publish Message (id=101) [test/1] 58 Publish Ack (id=2)</pre>
00b0 00c0	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 65 14.338086	Source 192.168.1.9 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7	54 8f 85 4a g. 7. The MQTT 239.255.255.250 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245	r traffic Protoc SSDP MQTT MQTT MQTT MQTT	-j protected by the TLS scheme. ol Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id=2) [test/1] 107 Publish Message (id=20] [test/1] 58 Publish Ack (id=2) 58 Publish Ack (id=20)
00b0 00c0 No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 65 14.388086 66 14.388992	88 c4 cb 08 56 67 51 Fig Source 192.168.1.9 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 192.168.1.7	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 3.122.42.245 3.122.42.245	Y traffic Protoc SSDP MQTT MQTT MQTT MQTT TCP	-j protected by the TLS scheme. ol Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id=2) [test/1] 107 Publish Message (id=101) [test/1] 58 Publish Ack (id=2) 58 Publish Ack (id=2) 58 Publish Ack (id=101) 54 S9738 + 3883 [ACK] Seq=141 Ack=73 Win=131840 Len=0
00b0 00c0 No.	6b b3 17 aa ad 5 f5 fa cb 13.604534 62 14.107147 63 14.337011 64 14.337011 65 14.338086 66 14.38892 67 14.643592	88 c4 cb 08 56 67 51 Fig Source 192.168.1.9 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245	54 8f 85 4a k g. 7. The MQTT 239.255.255.250 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 3.122.42.245 3.122.42.245 3.122.42.245	Y traffic Protoc SSDP MQTT MQTT MQTT MQTT TCP TCP	<pre>i protected by the TLS scheme. lengh Info l67 M-SEARCH * HTTP/1.1 l07 Publish Message (id-2) [test/1] l07 Publish Message (id-20) [test/1] 58 Publish Ack (id-2) 58 Publish Ack (id-20) 54 S9738 + 1883 [ACK] Seq=141 Ack=73 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=1277 Ack=57 Win=27136 Len=0</pre>
00b0 00c0 No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 65 14.338086 66 14.388992 67 14.643592 68 16.128595	88 c4 cb 08 56 67 51 Fig 92.168.1.9 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7	54 8f 85 4a k g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 3.122.42.245 3.122.42.245 3.122.42.245	VgQT traffic Protoc SSDP MQTT MQTT MQTT TCP TCP MQTT	-j protected by the TLS scheme. ol Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id-2) [test/1] 107 Publish Message (id-20) [test/1] 58 Publish Ack (id-20) 58 Publish Ack (id-20) 56 1883 + 59734 [ACK] Seq-141 Ack-73 Win-131840 Len=0 56 1883 + 59734 [ACK] Seq-177 Ack-57 Win-27136 Len=0 107 Publish Message (id=3) [test/1]
00b0 00c0	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 65 14.338086 66 14.388092 67 14.643592 67 14.643592 68 16.128505 69 16.379453	88 c4 cb 08 56 67 51 Fig Source 192.168.1.9 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245	54 8f 85 4a g. 7. The MQTT 239.255.255.250 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7	· · · VgQT · traffic Protoc SSDP MQTT MQTT MQTT TCP MQTT MQTT	<pre>i protected by the TLS scheme. left life left life</pre>
00b0 00c0	6b b3 17 aa ad 5 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 65 14.338086 66 14.388992 67 14.643592 68 16.128595 69 16.379453 70 16.379453	Source 192.168.1.9 192.168.1.9 192.168.1.9 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 192.168.1.7 192.168.1.7	· · · VgQT · traffic Protoc SSDP MQTT MQTT MQTT MQTT MQTT	<pre>i protected by the TLS scheme. length Info lf7 M-SEARCH * HTTP/1.1 lf7 Publish Message (id-2) [test/1] lf7 Publish Message (id-101) [test/1] lf8 Publish Ack (id-2) lf8 Publish Ack (id-2) lf8 Publish Ack (id-101) lf9 Publish Message (id-3) [test/1] lf9 Publish Message (id-3) [test/1] lf9 Publish Message (id-3) [test/1] lf9 Publish Ack (id-3) lfest/1] lf8 Publish Ack (id-3) lfest/1] lf8 Publish Ack (id-3) lfest/1] lf9 Publish Ack (id-3) lfest/1] lf</pre>
00b0 00c0	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 65 14.338086 66 14.388992 67 14.643592 68 16.128505 69 16.379453 70 16.380842	88 c4 cb 08 56 67 51 Fig 92.168.1.9 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 3.122.42.245 3.122.42.245 3.122.42.245 192.168.1.7 3.122.42.245	· · · VgQT · traffic Protoc SSDP MQTT MQTT MQTT TCP TCP MQTT MQTT MQTT MQTT MQTT	-j protected by the TLS scheme. ol Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id=2) [test/1] 107 Publish Message (id=101) [test/1] 58 Publish Ack (id=101) 54 59738 + 1883 [ACK] Seq=141 Ack=73 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=141 Ack=73 Win=27136 Len=0 107 Publish Message (id=3) [test/1] 107 Publish Message (id=151) [test/1] 58 Publish Ack (id=151) 58 Publish Ack (id=151)
00b0 00c0	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 64 14.337011 65 14.338086 66 14.388992 67 14.643592 67 14.643592 68 16.128595 69 16.379453 70 16.379453 71 6.380842 72 16.425611	Re c4 cb 08 56 67 51 Fig Source 192.168.1.9 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 3.122.42.245 3.122.42.245 3.122.42.245	54 8f 85 4a g. 7. The MQTT 239.255.255.250 3.122.42.245 192.168.1.7 3.122.42.245 3.122.42.245 3.122.42.245 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7	· · ·VgQT· · traffic Protoc SSDP MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT	<pre>-j protected by the TLS scheme. ol Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id-2) [test/1] 107 Publish Message (id-2) [test/1] 58 Publish Ack (id-2) 58 Publish Ack (id-2) 58 Publish Ack (id-2) 107 Publish Message (id-131) [test/1] 107 Publish Message (id-131) [test/1] 107 Publish Message (id-131) [test/1] 58 Publish Ack (id-3) 58 Publish Ack (id-3) 58 Publish Ack (id-3) 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=131840 Len=0 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=131840 Len=0 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=131840 Len=0 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=131840 Len=0 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=131840 Len=0 55 700 700 700 700 700 700 700 700 700 7</pre>
No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 65 14.338992 66 14.388992 67 14.643592 68 16.128505 69 16.379453 70 16.379453 71 16.380842 72 16.425611 73 16.591437	Source 192.168.1.9 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 3.122.42.245 3.122.42.245 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7	••••vgqT• • traffic • protoc SSDP MQTT MQTT MQTT MQTT MQTT MQTT MQTT TCP TCP	<pre>i protected by the TLS scheme. length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id-2) [test/1] 107 Publish Message (id-20) [test/1] 107 Publish Ack (id-2) 58 Publish Ack (id-2) 58 Publish Ack (id-10) 54 59738 + 1883 [ACK] Seq=141 Ack=73 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=141 Ack=73 Win=27136 Len=0 107 Publish Message (id-3) [test/1] 107 Publish Message (id-3) [test/1] 58 Publish Ack (id-3) 58 Publish Ack (id-3) 58 Publish Ack (id-3) 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=239 Ack=61 Win=27136 Len=0 </pre>
No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 64 14.337011 65 14.338086 66 14.388992 67 14.643592 68 16.128595 69 16.379453 70 16.379453 71 16.380842 72 16.425611 73 16.391437 74 17.094029	Source 192.168.1.9 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 Chongqin_91:f1:5	•••• •vgqT• • traffic • protoc SSDP MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT CP TCP • CP	<pre>-i protected by the TLS scheme. d Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id-20) [test/1] 107 Publish Message (id-101) [test/1] 58 Publish Ack (id-2) 58 Publish Ack (id-101) 54 59738 + 1883 [ACK] Seq=141 Ack=73 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=147 Ack=75 Win=27136 Len=0 107 Publish Message (id=3) 107 Publish Message (id=3) 107 Publish Ack (id=51) 58 Publish Ack (id=51) 58 Publish Ack (id=151) 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=194 Ack=77 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=294 Ack=77 Win=131840 Len=0 42 Who has 192.168.1.77 Tell 192.168.1.1</pre>
00b0 00c0	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 64 14.337011 65 14.338086 66 14.388992 67 14.643592 69 16.379453 70 16.379453 70 16.379453 71 6.388842 72 16.425611 73 16.591437 74 17.094629 Frame 68: 107 byt	Source 192.168.1.9 192.168.1.7 192.168.1 192.168	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 3.122.42.45 3.122.42.45 3.122.42.45 3.122.42.45 3.122.42.45 3.122.42.45 3.122.42.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.125 3.125 3.125 3.125 3.125 3.125 3.125	••••VgQT- • traffic SSDP MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTA MQT	<pre>.i protected by the TLS scheme. left line left lin</pre>
00b0 00c0 No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 65 14.338096 66 14.388992 68 16.128505 69 16.379453 71 16.380842 72 16.425611 73 16.591437 74 17.094029 Frame 68: 107 byt Frame 68: 107 byt	Source 192.168.1.9 192.168.1.7 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 192.1	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 3.122.42.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.122.45 3.125 3.125 3.125 3.125 3.125 3.125	traffic Protoc ssDP MQTT MQTT MQTT MQTT MQTT TCP TCP TCP MQTT MQTT MQTT MQTT f ARP i (856 bit st: zte_z	<pre>-j protected by the TLS scheme. ol Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id-2) [test/1] 107 Publish Message (id-2) [test/1] 107 Publish Ack (id-2) 58 Publish Ack (id-2) 58 Publish Ack (id-2) 58 Publish Ack (id-2) 107 Publish Message (id-3) [test/1] 107 Publish Message (id-151) [test/1] 108 Publish Ack (id-3) 109 Fublish Ack (id-3) 109 Fublish Ack (id-3) 100 fublish Message (id-151) [test/1] 107 Publish Message (id-151) [test/1] 107 Publish Message (id-151) 107 Publish Message (id-3) [test/3] 109 fublish Ack (id-3) 109 Fublish Ack (id-3) 109 fublish Ack (id-3) 109 fublish Ack (id-3) 100 fublish Message (id-3) [test/3] 100 fublish Ack (id-3) 100 fub</pre>
0060 00c0	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 64 14.337011 64 14.337011 65 14.338086 66 14.388992 67 14.643592 68 16.128505 69 16.379453 70 16.379453 71 16.380842 72 16.425611 73 16.391437 74 17.094029 Frame 68: 107 by Ethernet II, Srct Ethernet II, Srct	Source 192.168.1.9 192.168.1.9 192.168.1.7 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 21.168.1.7 3.122.42.245 21.168.1.7 3.122.42.245 21.168.1.7 3.122.42.245 21.168.1.7 3.122.42.245 21.168.1.7 3.122.42.245 21.168.1.7 3.122.42.245 21.168.1.7 3.122.42.245 21.168.1.	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 193.17	••• vgqT· • traffic Protoc SSDP MqTT MqTT MqTT MqTT TCP TCP MqTT MqTT<	<pre>-i protected by the TLS scheme. d Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id-20) [test/1] 107 Publish Message (id-20) [test/1] 58 Publish Ack (id-20) 58 Publish Ack (id-101) 54 59738 + 1883 [ACK] Seq=141 Ack=73 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=177 Ack=57 Win=27136 Len=0 107 Publish Message (id-31) [test/1] 107 Publish Message (id-31) [test/1] 58 Publish Ack (id-151) 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=194 Ack=77 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=204 Ack=77 Win=131840 Len=0 42 Who has 192.168.1.7? Tell 192.168.1.1 s) on interface \Device\WPF_{654F777B-4E6F-4784-BCE9-373F602802A7}, id 0 dice:8c (44:fb:5a:ad:ce:8c)</pre>
00b0 00c0 No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 64 14.337011 65 14.338086 66 14.388092 67 14.643592 67 14.643592 69 16.379453 70 16.379453 70 16.379453 71 6.591437 74 17.094029 Frame 68: 107 byt Ethernet II, Src: Internet Protocol Transmission Cont	Re c4 cb 08 56 67 51 Fig. Source 192.168.1.9 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 3.122.42.245 3.122.42.245 3.122.42.245 3.122.42.245 192.168.1.7 3.122.42.245 ttes on wire (856 bits : chongqin_91:f1:5f (1 Version 4, Src: 192 trol Protocol, Src Po	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 3.122.42.557 Chongqing1:f1:57 .573, Dat Port: 3.122.42 .572, Dat Port: 3.122 .572, Dat Port: 3.572, D	 YugqT. Ytraffic Protoc SSDP MQTT MQTT	<pre>-j protected by the TLS scheme. ol Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id-2) [test/1] 107 Publish Message (id-2) [test/1] 58 Publish Ack (id-2) 58 Publish Ack (id-2) 58 Publish Ack (id-101) 54 59738 + 1883 [ACK] Seq=141 Ack=73 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=141 Ack=75 Win=27136 Len=0 107 Publish Message (id-151) 107 Publish Message (id-151) [test/1] 58 Publish Ack (id-3) 58 Publish Ack (id-3) 58 Publish Ack (id-31) 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=230 Ack=61 Win=27136 Len=0 42 Who has 192.168.1.77 Te11 192.168.1.1 s) on interface \Device\WPF (654F7778-4E6F-4784-BCE9-373F602802A7}, id 0 dtce:8c (44:fb:5a:ad:ce:8c) 1 2141, Ack: 73, Len: 53 </pre>
00b0 00c0 No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 64 14.337011 64 14.33702 71 16.380842 66 16.128505 69 16.379453 71 16.380842 72 16.425611 73 16.591437 74 17.094029 Frame 68: 107 by Frame 68: 107 by Transmission Cont V Telemetry Transmission Cont	Source 192.168.1.9 192.168.1.9 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 1.12.42.245 192.168.1.7 1.12.42.245 192.168.1.7 1.12.42.245 192.168.1.7 1.12.42.165 192.168.1.7 1.12.42.165 192.168.1.7 1.12.42.165 192.168.1.7 1.12.42.165 192.168.1.7 1.12.42.165 192.168.1.7 1.12.42.165 192.168.1.7 1.12.42.165 192.168.1.7 1.12.42.245 192.168.1.7 1.12.42.245 192.168.1.7 1.12.42.245 192.168.1.7 1.12.42.245 192.168.1.7 1.12.42.245 192.168.1.7 1.12.42.245 192.168.1.7	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 3.122.42.45 3.122.42.45 3.122.42.45 3.125 3.125 3.125 3.125 3.125	 YegqT. Ytraffic Protoc SDP MQTT MQTT	<pre>-i protected by the TLS scheme. ol Length Info 167 M-SEARCH * HTTP/1.1 167 Publish Message (id=2) [test/1] 107 Publish Message (id=2) [test/1] 107 Publish Message (id=2) [test/1] 58 Publish Ack (id=2) 58 Publish Ack (id=2) 56 1883 + 59734 [ACK] Seq=141 Ack=73 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=177 Ack=57 Win=27136 Len=0 107 Publish Message (id=3) [test/1] 107 Publish Message (id=3) [test/1] 107 Publish Ack (id=151) 58 Publish Ack (id=151) 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=194 Ack=77 Win=27136 Len=0 42 Who has 192.168.1.77 Tell 192.168.1.1 s) on interface \Device\MPF_{654F777B-4E6F-47B4-BCE9-373F602802A7}, id 0 dice:8c (44:fb:5a:adice:8c) 1; 141, Ack: 73, Len: 53</pre>
00b0 00C0	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 64 14.337011 64 14.337011 65 14.338086 66 14.388992 67 14.643592 69 16.379453 70 16.379453 70 16.380842 72 16.425611 73 16.391437 74 17.094029 Frame 68: 107 byt Ethernet II, Srcc Transmission Cont 40 Telemetry Tran	Source 192.168.1.9 192.168.1.7 3.122.42.245 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 22.168.1.7 3.122.42.245 22.168.1.7 3.122.42.245 22.168.1.7 3.122.42.245 22.168.1.7 3.122.42.245 zte_ad:ce:8c	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 193.17	 · · ·vgqT· · traffic Protoc SSDP MQTT MQTT	<pre>-i protected by the TLS scheme. d Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id-20) [test/1] 107 Publish Message (id-210) [test/1] 58 Publish Ack (id-20) 58 Publish Ack (id-101) 54 59738 + 1883 [ACK] Seq=147 Ack=73 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=177 Ack=57 Win=27136 Len=0 107 Publish Message (id-31) [test/1] 107 Publish Message (id-31) [test/1] 58 Publish Ack (id-151) 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=2194 Ack=77 Win=27136 Len=0 42 Who has 192.168.1.7? Tell 192.168.1.1 s) on interface \Device\WPF_[654F777B-4E6F-4784-BCE9-373F602802A7], id 0 dice:8c (44:fb:5a:ad:ce:8c) 1: 141, Ack: 73, Len: 53 </pre>
00b0 00C0	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 64 14.337011 64 14.337011 65 14.388082 66 14.388092 67 14.643592 67 14.643592 69 16.379453 70 16.379453 70 16.388042 72 16.42561 73 16.591437 74 17.094029 Frame 68: 107 byt Ethernet II, Src Internet Protocol Transmission Cont %Q Telemetry Trans 0 44 fb 5a ad co	Source 192.168.1.9 192.168.1.7 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 192.168.1.7 192.168.1.7	54 8f 85 4a k g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 192	 vgqT. traffic Protoc SSDP MqTT MqTT	<pre>.'i protected by the TLS scheme. ol Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id-2) [test/1] 107 Publish Message (id-2) [test/1] 58 Publish Ack (id-2) 58 Publish Ack (id-2) 58 Publish Ack (id-2) 58 Publish Ack (id-2) 107 Publish Message (id-11) [test/1] 107 Publish Message (id-151) [test/1] 107 Publish Message (id-151) [test/1] 58 Publish Ack (id-3) 59 Publish Ack (id-151) 54 59738 + 1883 [ACK] Seq=134 Ack=77 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=230 Ack=61 Win=27136 Len=0 42 Who has 192.168.1.77 Tell 192.168.1.1 ts) on interface \Device\MPF_(654F7778-4E6F-4784-8CE9-373F602802A7}, id 0 dice:8c (44:fb:5a:ad:ce:8c) t 141, Ack: 73, Len: 53 Fucrynted content of </pre>
00b0 00c0 No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 64 14.337011 64 14.337011 65 14.338086 66 14.38086 67 14.643592 68 16.128505 69 16.379453 71 16.380842 72 16.425611 73 16.591437 74 17.094829 Frame 68: 107 by Transmission Conf W2 Telement J, Srct Transmission Conf 00 44 fb 5a ad c 00 95 d9 6 13 4	Source 192.168.1.9 192.168.1.9 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.545 192.168.1.7 3.122.42.545 192.168.1.7 3.122.42.545 192.168.1.7 3.122.42.75 2te_ad:ce:8c tes on wire (856 bits: 192.168.1.7 3.127.42.75 2te_ad:ce:8c tes on wire (756 bits: 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 1	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 192.168.1.7 192.168.1.7 Chongqin_91:f1:57) , 107 bytes capturece 8:61:37:91:f1:57) , 107 bytes capturece 8:61:37:91:f1:57) , 107 bytes capturece 5:67:83:91:f1:57) , 107 bytes capturece 5:67:83:91:57:83, Dst Port: 1:68.1.7 Dst: 3.122. 1:68.1.7 Dst: 3.122. 1:69.1.8 Dst Port: 1:69.1.8 Dst	 · · · vgqT· ' traffic Protoc SDP MQTT MQTT	<pre>-i protected by the TLS scheme. d Length Info 167 M-SEARCH * HTTP/1.1 167 Publish Message (id-2) [test/1] 107 Publish Message (id-2) [test/1] 58 Publish Ack (id-2) 58 Publish Ack (id-2) 54 59738 + 1883 [ACK] Seq=141 Ack=73 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=177 Ack=57 Win=27136 Len=0 107 Publish Message (id-31) [test/1] 107 Publish Message (id-31) [test/1] 58 Publish Ack (id-151) 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=27136 Len=0 42 Who has 192.168.1.77 Tell 192.168.1.1 s) on interface \Device\WFF_{654F777B-4E6F-4784-BCE9-373F602802A7}, id 0 dice:8c (44:fb:5a:ad:ce:8c) 1: 141, Ack: 73, Len: 53 Encrypted content of </pre>
00b0 00C0 No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 64 14.337011 64 14.337011 65 14.337011 64 14.337011 64 14.337011 65 14.338086 66 14.38092 67 14.643592 68 16.128505 69 16.379453 70 16.379453 71 16.380842 72 16.425611 73 16.591437 74 17.094029 Frame 68: 107 byt Ethernet II, Srcc Thremet Protoco Transission Comi 70 Telemetry Transiston Comis 70 4 4f b5 a adc 71 6 49 54 60 61 34 72 7 55 60 61 34	Source 192.168.1.9 192.168.1.7 3.122.42.245 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 zte_ad:ce:8c tes on wire (856 bits: : Chongdin_91:fl:5f (i 1 Version 4, Src: 192 trobsport Protocol, Src Ponsport Protocol, Sup 1 fl 08 08 06 36 08 19 fl 08 08 06 80 68 45 ac 09 75 bd 06 64 ac 65 ce	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 192.17 192.17 192.17 192.1	 · · · vgqT· ' traffic Protoc SSDP MQTT MQT	<pre>-i protected by the TLS scheme. ol Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id-20) [test/1] 107 Publish Message (id-20) [test/1] 58 Publish Ack (id-20) 58 Publish Ack (id-101) 54 59738 + 1883 [ACK] Seq=147 Ack=73 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=147 Ack=75 Win=27136 Len=0 107 Publish Message (id-31) [test/1] 58 Publish Ack (id-3) 58 Publish Ack (id-451) 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=194 Ack=77 Win=27136 Len=0 42 Who has 192.168.1.7? Tell 192.168.1.1 is) on interface \Device\WPF_(654F777B-4E6F-47B4-BCE9-373F602802A7], id 0 dice:8c (44:fb:5a:ad:ce:8c) : 141, Ack: 73, Len: 53 Encrypted content of the MQTT message </pre>
00b0 00c0 No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 65 14.338092 66 14.388992 67 14.643592 68 16.128505 69 16.379453 71 16.591437 72 16.425611 73 16.591437 74 17.094029 Frame 68: 107 by Ethernet II, Src Internet Protocoo 00 5d 66 134 02 a f5 e 5 a 6 09 5d 66 134 02 a f5 e 5 a 6 08 Q2 03 83 f3 Q	Source Fig 192.168.1.9 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 192.168.1.7 192.168.1.7 192.	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 192.17 192.188.1 192.	•••vgqT· ' traffic Protoc SSDP MQTT	-i protected by the TLS scheme. ol Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id-2) [test/1] 107 Publish Acs (id-2) [test/1] 108 Publish Ack (id-20) 58 Publish Ack (id-20) 109 Publish Message (id-101) [test/1] 109 Publish Message (id-110) [test/1] 109 Publish Message (id-110) [test/1] 109 Publish Message (id-110) [test/1] 109 Publish Ack (id-3) 109 Publish Ack
00b0 90c0 No.	6b b3 17 aa ad 9 f5 fa cb Time 61 13.604534 62 14.107147 63 14.337011 64 14.337011 64 14.337011 64 14.337011 65 14.338080 66 14.383992 67 14.643592 68 16.128505 69 16.379453 71 16.380842 72 16.425611 73 16.591437 74 17.094429 Frame 68: 107 by Frame 68: 107 by 00 44 fb 5a ad c 00 05 do 13 4 01 00 53 45 e9 5a 01 00 20 33 35 44 85	Source 192.168.1.9 192.168.1.9 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7	54 8f 85 4a g. 7. The MQTT Destination 239.255.255.250 3.122.42.245 192.168.1.7 192.168.1.7 192.168.1.7 3.122.42.245 192.168.1.7 3.122.42.245 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 192.168.1.7 Chongqin_91:f1:57), 107 bytes capturece 8:6f3:89:171:f1:57) 5f 06 00 45 00 5f 06 00 45 00 57 37 42 f31 65 31 73 75 70 31 37 37 43 34 34 37 73 43 34 34 37 43 34 35 34 3	· · · vgqT· · traffic Protoc SDP MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT MQTT 4. (856 bit) Sci 2te 2, 42. 1883, Sec · · · · · · · · · · · · · · · · · · ·	<pre>-i protected by the TLS scheme. d Length Info 167 M-SEARCH * HTTP/1.1 107 Publish Message (id=101) [test/1] 107 Publish Message (id=101) [test/1] 58 Publish Ack (id=2) 58 Publish Ack (id=101) 54 59738 + 1883 [ACK] Seq=171 Ack=73 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=172 Ack=57 Win=27136 Len=0 107 Publish Message (id=151) [test/1] 58 Publish Ack (id=151) 54 59738 + 1883 [ACK] Seq=194 Ack=77 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=194 Ack=77 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=194 Ack=77 Win=131840 Len=0 56 1883 + 59734 [ACK] Seq=194 Ack=77 Win=131840 Len=0 42 Who has 192.168.1.77 Tell 192.168.1.1 s) on interface \Device\WFF (654F7778-4E6F-4784-8CE9-373F602802A7), id 0 dice:8c (44:fb:5a:ad:ce:8c) 1: 141, Ack: 73, Len: 53 Encrypted content of the MQTT message </pre>

Fig. 8. The MQTT traffic protected by the AES scheme.

to the tiny differences of around 1 - 2 milliseconds. If those computational delays are accumulated with the communication delays, then there is no significant change in the total communication and computational delays. As for plaintext and TLS schemes, there is no additional computation delay because the MQTT communication process implemented in Python programming language basically handles plaintext data for both plaintext and TLS communication schemes. In addition, the TLS feature in this MQTT communication scheme is executed by the lower layer process (network kernel).

The results of network cost from 1 to 30 sessions for those three communication schemes are presented in Fig. 12. The network cost of the TLS scheme is the highest one among all schemes in all sessions conducted in this experiment. These results are obvious because, the TLS scheme should conduct a TLS handshake including certificate exchange and authenticated key



Fig. 9. Communication delays for plaintext, AES, and TLS schemes.

exchange among MQTT publisher, broker, and subscriber prior to having MQTT communications. As a result, the TLS communication scheme generates a greater number of IP packets for securing the MQTT communications.



Fig. 10. Frame sizes for all communication schemes.



Fig. 11. Additional computational delays for three communication schemes and three payload sizes.

Eventually, based on the results in Fig. 7 to Fig. 12, the TLS communication scheme is proven to be secure, but it has the highest cost among all schemes in terms of communication delay and the number of packets in the network. Instead of using the TLS scheme, the AES cryptography-based communication scheme is preferable for devices with low computational resources considering the communication delay and network cost, which are not significantly differ with respect to the plaintext-based communication scheme in the MQTT communication. However, the use of AES-cryptography in the MQTT communication should be followed by session key usage by means of authenticated key exchange or session key scheduling so that the attackers cannot crack the security of the MQTT communication easily.



Fig. 12. Network cost for Plaintext, AES, and TLS communication schemes.

IV. DISCUSSION

Confidentiality of the MQTT messaging is a crucial aspect of hosting reliable and secure services in the IoT. One of the discussions in hardening the security in the MQTT is utilizing the TLS approach, which is proven to be the most secure and reliable support system for internet and mobile banking transactions. However, the use of the TLS approach in IoT devices is problematic because the TLS may degrade the QoS of IoT devices with low computational resources.

Instead of using the TLS, the widely implemented AES cryptography is more appropriate for IoT devices with low computational resources. This claim is valid because it is supported by the experimental results in the previous section, which demonstrate the excellent performance of the AES approach with respect to the TLS approach in terms of network cost, frame size, and communication delay.

In addition to the AES approach, the use of lightweight-AKE methods is encouraged for increasing the level of security by dynamically changing secret keys. If by any chance the previous or current session key is successfully guessed by attackers, then the following session key is still considered to be safe within the limited time of the session key usage because attackers need some number of times to correctly guess the key. Furthermore, as long as the session time is shorter than the attackers' computation time for guessing the correct key, then the confidentiality of the information can be guaranteed during the MQTT communications.

V. CONCLUSION

The MQTT protocol is commonly known as the protocol for supporting devices-to-devices communications in the Internet of Things (IoT) environment. However, the MQTT protocol is not protected with a certain security mechanism except for using the TLS in the lower layer. As a consequence, MQTT communication becomes slower and burdensome for the network due to generating a greater number of IP packets. This research investigates the use of the AES cryptography-based communication scheme against the TLS-based communication scheme for providing end-to-end secure communication channels from the MQTT publishers to the MQTT subscribers. Experimental results show that the TLSbased communication scheme possesses the highest cost in terms of communication delay and network cost among all schemes in the experiment. Eventually, the AES-based MQTT communication scheme is more beneficial for the MQTT communication scheme because its communication delay and network cost are considered equal to the basic MQTT communication scheme that sends data in plaintext mode.

ACKNOWLEDGEMENT

This paper is part of the research project collaboration between School of Electrical Engineering Telkom University and Department of Architecture Engineering Diponegoro University about Secure Communication in Smart Mosque in which funded through matching fund program.

REFERENCES

- C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog computing for sustainable smart cities: A survey," ACM Comput. Surv., vol. 50, 2017.
- [2] B. Mishra and A. Kertesz, "The use of MQTT in M2M and IoT systems: A survey," *IEEE Access*, vol. 8, pp. 201071–201086, 2020.
- [3] S. Seneviratne, Y. Hu, T. Nguyen, G. Lan, S. Khalifa, K. Thilakarathna, M. Hassan, and A. Seneviratne, "A survey of wearable devices and challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2573–2620, 2017.
- [4] A. A. Wardana and R. S. Perdana, "Access control on internet of things based on publish/subscribe using authentication server and secure protocol," in 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), 2018.
- [5] G. Vrettos, E. Logaras, and E. Kalligeros, "Towards standardization of MQTT-alert-based sensor networks: Protocol structures formalization and low-end node security," in 2018 IEEE 13th International Symposium on Industrial Embedded Systems (SIES), 2018.
- [6] S. Shin and K. Kobara, "Efficient augmented password-only authentication and key exchange for IKEv2," RFC Editor, 2012.
- [7] S. Shin, K. Kobara, C.-C. Chuang, and W. Huang, "A security framework for MQTT," in 2016 IEEE Conference on Communications and Network Security (CNS), 2016.
- [8] M. Calabretta, R. Pecori, M. Vecchio, and L. Veltri, "MQTT-Auth: a token-based solutionto endow MQTT with authentication and authorization capabilities," *Journal of Communications Software and Systems*, vol. 14, pp. 320–331, 2018.
- [9] A. Velinov, A. Mileva, S. Wendzel, and W. Mazurczyk, "Covert channels in the MQTT-based internet of things," *IEEE Access*, vol. 7, pp. 161899–161915, 2019.
- [10] U. Arom-oon, "An AES cryptosystem for small scale network," in 2017 Third Asian Conference on Defence Technology (ACDT), 2017.
- [11] M. O'Neill, S. Ruoti, K. Seamons, and D. Zappala, "TLS inspection: How often and who cares?," *IEEE Internet Computing*, vol. 21, no. 3, pp. 22–29, 2017.
- [12] F. Dewanta, "Secure microservices deployment for fog computing services in a remote office," in 2020 3rd International Conference on Information and Communications Technology (ICOIACT), 2020.
- [13] X. Liu, T. Zhang, N. Hu, P. Zhang, and Y. Zhang, "The method of internet of things access and network communication based on MQTT," *Computer Communications*, vol. 153, pp. 169–176, 2020.
- [14] A. Liu, A. Alqazzaz, H. Ming, and B. Dharmalingam, "Iotverif: Automatic verification of SSL/TLS certificate for IoT applications," *IEEE Access*, vol. 9, pp. 27038–27050, 2021.
- [15] J. Li, R. Chen, J. Su, X. Huang, and X. Wang, "ME-TLS: Middlebox-enhanced TLS for internet-of-things devices," *IEEE Internet of Things Journal*, vol. 7, pp. 1216–1229, 2020.

- [16] A. Badholia, V. Verma, and S. K. Kashyap, "Wep, Wap and Wap2 wireless network security protocol: A compact algorithm: (wireless network security protocol)," in 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2019.
- [17] S. Jain, S. Pruthi, V. Yadav, and K. Sharma, "Penetration testing of wireless encryption protocols," in 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), 2022.
- [18] J.-D. Wu, Y.-M. Tseng and S.-S. Huang, "An identity-based authenticated key exchange protocol resilient to continuous key leakage," *IEEE Systems Journal*, vol. 13, pp. 3968–3979, 2019.
- [19] S. Akleylek and K. Seyhan, "A probably secure Bi-GISIS based modified AKE scheme with reusable keys," *IEEE Access*, vol. 8, pp. 26210–26222, 2020.
- [20] R. J. Coppen, "MQTT Version 3.1.1," 29 October 2014. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqttv3.1.1-os.html. [Accessed 11 August 2022].