On The Possibilities of Cybercrime in IoT Devices

Yosafat Marselino Agus[†], Muhammad Dzakwan Falih[‡], Gandeva Bayu Satrya[‡*]
[†]School of Electrical Engineering, Telkom University, Republic of Indonesia
[‡]School of Applied Science, Telkom University, Republic of Indonesia
[†]ninoagus@protonmail.com, [‡]dzakwan@ieee.org, [‡*]gbs@telkomuniversity.ac.id

***Abstract*—Internet of Things (IoT) has been augmenting the emerging technologies and certainly been varying our daily life. The adoption of this technology is strengthened by the growth of connecting devices as shown in recent literature. However, responsibility related to secure communication also needs to increase as the number of connections grows. For instance, cybercrime might happen if simple topology and protocol are not implemented on IoT applications, or the communications from sensors to the Internet are weakly defined. This research reviews the vulnerability of existing topology and configuration on IoT. A secure communication is proposed between sensor nodes and the Internet. Further, this research demonstrates the feasibility of recommended protocol communication for several IoT devices through real testbed for smart home.**

***Index Terms*—IoT, protocol, sensor nodes, secured communication, vulnerability.**

I. Introduction

Communication between smartphone devices or sending commands to home devices is no longer a fantasy with the development of future wireless communication technology. Data communication from machine to machine or human to machine using Internet connection is called as Internet of Things (IoT). IoT is one of the most important IT tendencies and it is getting more appealing in home or factory automation. Figure. 1 illustrates an IoT communication architecture consists of sensors, smart objects, smart devices, gateways, back-end data centers, and services [1].

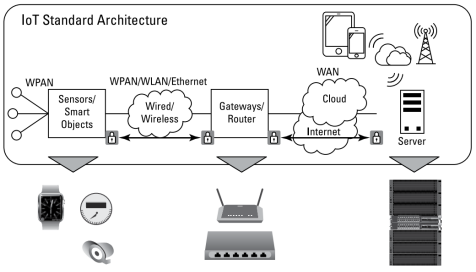


Figure. 1: IoT system architecture

Data security and privacy certainly become the main topics for any organization that is connected to the Internet. On February 2018, the Interpol National Central Bureau held a meeting in Vienna where 43 cybercrime investigators and digital forensics experts were invited from 23 countries to investigate cyberattack simulations on IoT devices [2]. They have given four guidance on how to secure connected devices from cybercrime. The risks are far greater on IoT because of much more personal and important data collected from abundant IoT devices that could be used for vicious intentions. Thus, providing a suitable security control or communication protocol for consumers has become a necessity for IoT device vendors.

IoT needs to incorporate various sensor nodes or smart objects, computer or server and wireless connection, which are using different communication protocols. Based on TCP/IP protocol stack, several communication protocols have been proposed on each layer. For this research, it will focus on application layer. Observing last recent research update related to application's protocol for IoT, five outstanding protocols are HTTP [3], XMPP [4], MQTT [5], AMQP [6], and CoAP [7]. However, MQTT is the most candidates for standard IoT communication protocols due to it reduces protocol overheads and provides reliable delivery of messages [8]–[10].

Researches in recent years focusing on IoT security problems aimed for different aspect. Commencing with the survey by [11]] that proposed security requirements specific to IoT systems by taking into consideration network security, identity management, privacy, trust, and resilience from well-known standardization technology. Another approach from [12] designed and implemented a reliable message transmission system based on MQTT Protocol for IoT devices. However, [13] proposed a simple security framework for MQTT (for short, AugMQTT) by incorporating the AugPAKE protocol that gives security against passive attacks, active attacks, and off-line dictionary attacks. In the era of urban computing and IoT systems, [14] identified three prominent approaches to provide seamless and lightweight IoT interactions, highlighting a representative, standardized protocol for each of these approaches. Acknowledging those literature, there are still many available research areas that can be explored in order to provide optimized topology, secure scheme, or modest algorithm for IoT architecture.

Initiating with the deployment MQTT protocol for IoT adopted from [10], this research shows that vulnerability i.e. stealing data information can give a serious hit. That reason brought this research to optimization of MQTT protocol implementation in IoT environment. By using that topology, the pen-testing for deep investigations were successfully conducted and developed secure end to end communication from sensor nodes to back-end data centers. The implementation and testing were carried out using three Raspberry Pi 3B as IoT devices, Mosquitto as MQTT broker, Eclipse Paho library on Python for IoT, and Wireshark for protocol analyzer.

The contributions of this research are as follows:

- Proposing a secure topology by using MQTT over web-socket where the unregistered IoT devices might not connect.
- Providing an optimized MQTT protocol configuration based on the testbed by adding lightweight cryptography from physical layer to application layer.
- Presenting a guidance for secure IoT device topology by considering smart hospital, smart home, smart factory, or other enhanced IoT technology serving vital information.

The section II reviews existing works in IoT security. While section III explains the proposed scheme for secure end to end communication among IoT devices. Then, the details of unsecure and secure experiments are provided in section IV. Finally, section V gives the conclusions of this research.

II. Related Works

Singh et al. proposed a secure version of MQTT and MQTT-SN protocols *(SMQTT and SMQTT-SN)* in which the security feature is augmented to the existing MQTT protocol based on Key/Ciphertext Policy-Attribute Based Encryption (KP/CP-ABE) using lightweight Elliptic Curve Cryptography [15]. According to the theoretical and experimental analysis, the proposed schemes showed better performance than the scheme described in [3] which was based on KP/CP-ABE. However, those proposed schemes were just simulation-based and has not yet been implemented on a real IoT platform.

Dragomir et al. have done surveys in the last several years to prevent attackers from obtaining control over the IoT devices [11]. Many solutions have been proposed by IoT security researches in the last years, but most of them were not standardized or interoperable. The research evaluated security requirements specific to IoT systems considering the network security, identity management, privacy, trust, and resilience. The results were summarized by functionality and security capabilities and specified by standardization bodies e.g., IETF, IEEE, or industry alliances.

Andy et al. discussed several reasons of many IoT system that does not implement adequate security mechanism. The research also demonstrated and analyzed how to easily attack this protocol using several attack scenarios [16]. Considering Shodan as the problem identification, it showed that *24998 brokers* who have connection code of *"0"* were easier to be attacked because this kind of broker did not use any client authentication mechanism. To the best of authors' knowledge, this paper just concluded with a suggestion that MQTT protocol must be implemented e.g., TLS, ECC, or other recent secured protocol on IoT devices.

Grgić et al. [17] proposed a web-based IoT solution aimed for monitoring, tracking and analyzing data in agriculture area. The research was done by installing and connecting sensors to devices (e.g. Arduino Uno, Raspberry Pi) which main task was to send data (i.e. temperature and moisture values) to a central server over MQTT. The results showed that MQTT protocol has several characteristics i.e., low overhead, asynchronous communication, low complexity and low power.

Niruntasukrat et al. [18] presented design and implementation of an authorization mechanism by using MQTT for IoT. The design was based on the OAuth 1.0a, an open authorization standard for web applications. Some redesign and modification have been made to the base framework to make it fit within the MQTT environment. However, the authorization mechanism demanded two sets of credentials for the device to access the MQTT broker. Through experiments and real services, the proposed authorization process worked as intended, and the incurred overhead did not affect user's experiences.

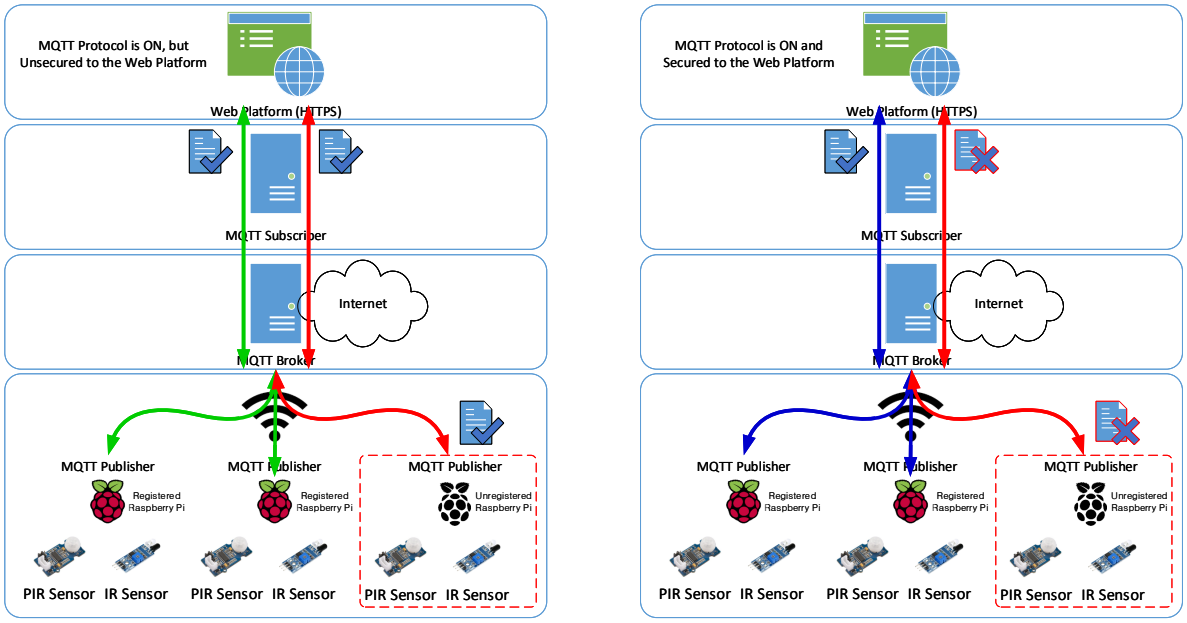
III. Proposed Framework

For the sake of completeness in securing communication between the IoT devices, this research implements MQTT as the prime candidate from recent existing literature. Figure. 2a shows the deployment of conventional MQTT implementation for IoT topology or called as unsecured transaction framework [10]. The next section provides a deep investigation for this topology. To the best of authors' knowledge, most of the simulation and evaluation of MQTT just gave assurance on the protocol level from sensor nodes to MQTT broker. But from adversary's point of view *(unregistered sensor nodes)*, the data sensor over web-socket still can be diagnosed with plain text. For concealing this issue, this research suggests adopting auto-keying between publishers and subscribers to reject the unregistered sensor nodes. Proposed IoT topology for securing end to end communication from sensor nodes to web platform (HTTPS) or called secured transaction framework is depicted in Figure. 2b.

A. Unsecured Transaction Framework

The implementation was deployed using two unit of Raspberry Pi 3B as registered devices and one unit as an unregistered device. The Wi-Fi connection had medium access to MQTT Publisher. The configuration for MQTT broker, MQTT subscriber, and database server had been built by following the MQTT library or documentation as can be seen in Figure. 2a. This topology also provided HTTPS web server for pooling the data's sensor. The pen-testing this topology are discussed in the following section. MQTT is a topic-based publish/subscribe protocol. Every message is published on a designated topic, and every subscription has a topic filter that can include wildcards. So, the authorization is in terms of publishing/subscribing and topic designating.

The most common problem in implementing IoT system based on MQTT protocol is the poorly configured devices. The default configuration of MQTT devices running on Mosquito service and Paho MQTT library on Python enables the data with unencrypted format to be transmitted over the Internet. By imitating the format data and the topic, some adversaries may spoof the broker's and the subscriber's sides. Faking the data or crippling the server is possible in this scenario. Upon this hypothesis, further analysis will be addressed in the next section.



(a) Conventional IoT topology using MQTT [10]

(b) Proposed secure web-socket IoT topology

Figure. 2: Comparison IoT architecture with optimized configuration by using MQTT protocol

B. Secured Transaction Framework

Slightly, there is no difference between the proposed topology and the conventional IoT topology. As shown in Figure. 2b, this research recommends lightweight cryptography algorithm for IoT devices to camouflage the messages from sensor node to web platform or to prevent the possibility of the publisher's data stealing *(unregistered Raspberry Pi device)*. The implementation of costumed Fernet library at the publisher's, broker's, subscriber's, and web platform's sides, the secure topology were enforced to be more secure.

Fernet cryptography guarantees that a message, that was encrypted by using it, cannot be manipulated or read without the key. Fernet is categorized a symmetric authenticated cryptography. This research employs 128-bit AES and HMAC for the authentication and modifies the Fernet to be more robust in key rotation. Thus, the IoT devices were authenticated and only the authorized devices with the key can publish and subscribe to a restricted topic. This research hypothesizes that the proposed topology could give protection from fake publishers, fake devices, and invalid uses of data.

Algorithm 1 Autokeying Publisher Encryption

```
INITIALISE publisher_ID;
INITIALISE dataSensors_sent;
INITIALISE MQTT_connection;
passcode = HMAC "encoded_password_string";
final_key = derive (passcode);
while true do
    ENCODE dataSensors_sent
    PUBLISH dαtαSεnsϕrs_şęñť                ▷ encrypted data
end while
```

Algorithm 2 Autokeying Subscriber Decryption

```
INITIALISE subscriber_ID;
INITIALISE MQTT_connection;
INITIALISE final_key;
final_key = generated_key_publisher;        ▷ from publishers
while true do
    SUBSCRIBE dαtαSεnsϕrs_şęñť
    DECODE dαtαSεnsϕrs_şęñť                 ▷ using final_key
end while
```

1) Publisher Encryption Algorithm: The publisher's identity that send data *(dataSensors_sent)* is known. The publisher will forward the data to the MQTT broker by using MQTT connection. Upon sending a secret key to one connection, the secret key will do HMAC for each string of passwords that will be used on the subscriber's side as a guarantee of information integrity. The same key generation method can be used by all publishers, so the subscriber can decode all of the incoming information with only one key. The next process with the key is to encrypt the data *(dαtαSεnsϕrs_şęñť)* sent by the publisher as presented in Algorithm 1.

2) Subscriber Decryption Algorithm: It is known that the identity of which subscriber will receive the encrypted data from the publisher. Indeed, the connection between the subscriber and the MQTT broker has been established and the key of the publisher *(final_key)* has been generated as presented in Algorithm 2. Then, the incoming key will be decrypted by using the key. If the format of incoming data could not be recognized, then the subscriber will reject the received data from publisher. Moreover, the subscriber will get encrypted data *(dαtαSεnsϕrs_şęñť)* which will then be decrypted to be displayed on the web platform.

IV. Security Analysis

A. Unsecured Transaction Experiment

Figures. 3 and 4 shows the results of network analysis with topology as in section III is successfully extracted. Certainly, MQTT has been implemented properly which can be seen from the Wireshark output below. Data from sensor node will be sent to the web platform *(HTTPS)*. Although the broker and MQTT broker are secure, the data sent via web-socket still can be seen *(plain text)*. Therefore, it can be concluded that it is not enough for guaranteeing the data's integrity only by implementing the publisher node, MQTT broker, and subscriber node. Next, the solution to this problem will be explained.

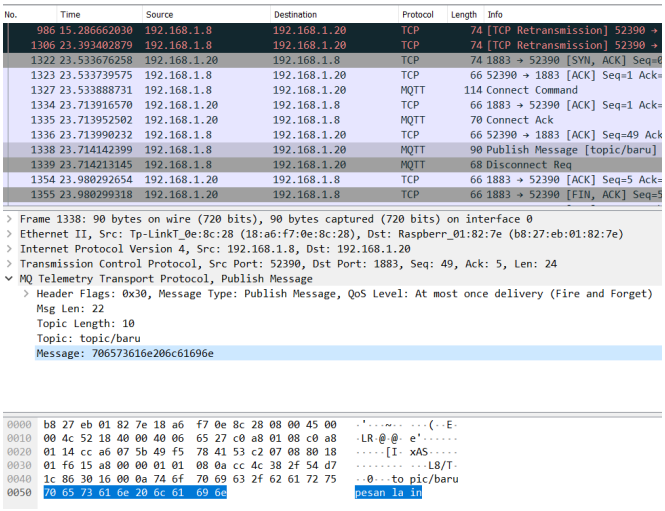


Figure. 3: Analyzing for the conventional topology

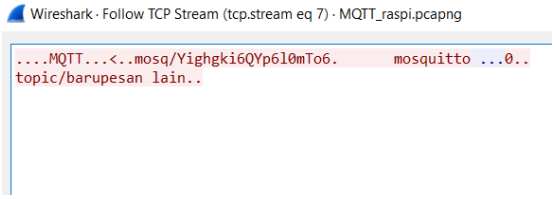


Figure. 4: Captured packets from the conventional topology

B. Secured Transaction Experiment

As seen in Figures. 5 and 6, the proposed IoT topology is more secure in terms of data privacy. The *dataSensor_sent* from the publisher was already authenticated and encrypted by using base-64 format. Though Fernet cryptography is symmetric encryption method, this problem was addressed by using *generate_key* to keep rotating the secret key for each data released by the publisher.

To the best of authors' knowledge, it can also be seen in Figure. 6 that the data are no longer recognized as a plain text but a cipher text which can also eliminate the possibility of attacks in data integrity category.

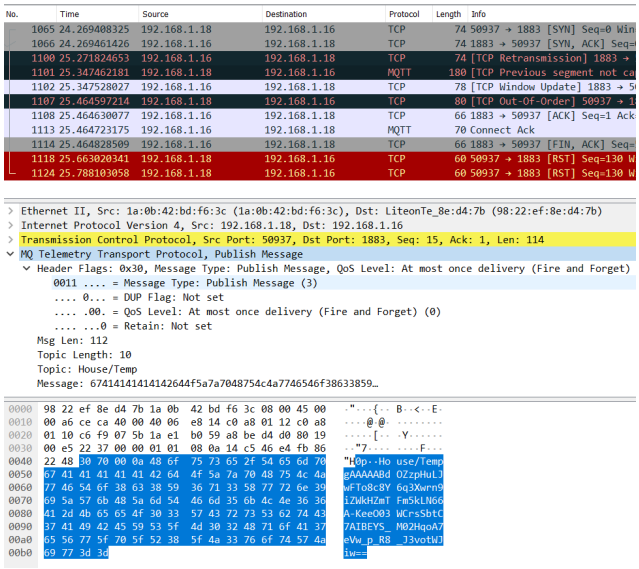


Figure. 5: Analyzing for the secured topology

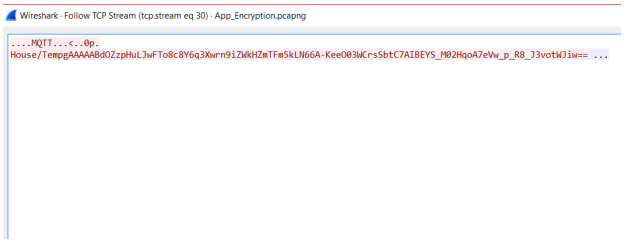


Figure. 6: Captured packets from the secured topology

C. Discussion of Experiment

The proposed secure web-socket topology was indeed proved to be twice bigger than the conventional topology by examining data length during the communication between publisher and subscriber. Upon the product specification of Raspberry Pi 3 Model B, the cryptography computation of encrypted data for 180 bytes (as depicted in Figure. 5) still can be addressed by Quad Core 1.2GHz Broadcom BCM2837 64bit CPU. To the best of authors' knowledge, this proposed topology should be implemented for future secure end to end communication of IoT devices.

Based on the two experiment categories, the basic configuration or features provided by vendors should be given more attention during the deployment of IoT devices. With the increasing number of sensor nodes that will serve the consumers, it is very important to ensure the security of the network e.g., optimizing the configuration, pen-testing the networks, and implementing third party authentication. This research proposes auto-keying either encryption or decryption in ensuring integrity of the passing data (as shown in Figure. 6). In accordance with the previous explanation, it is proven that the proposed IoT topology is more secure compared to conventional IoT topology which only uses MQTT protocol.

V. Conclusion

In this research, the feasibility of MQTT were implemented and analyzed to enable secure communication for IoT devices with real testbed. However, it also showed the drawback of MQTT protocol when the publisher sends the topic to subscriber throughout web platform. As recommendation, this research proposed a secure web-socket IoT topology between publisher and subscriber combined with MQTT protocol for IoT devices. Moreover, this research also suggested the use of encryption and decryption to enhance security of end to end communication in the implementation of smart hospital, smart home, smart factory, etc. Further security analysis of secure web-socket for IoT devices under different attack scenarios might be considered. This tesbed research was based on the internal threat, future research needs to scrutinize the external threat and use different vendor IoT devices.

Acknowledgment

This research was collaborated between School of Applied Science and School of Electrical Engineering, Telkom University. This research was also funded by PPM, Telkom University.

References

[1] Lawrence Miller. Iot security for dummies, carrie a. johnson, ed, 2016.

[2] Interpol. Digital security challenge. https://tinyurl.com/yyjz6g2r, February 2018. Accessed: 2019-07-07.

[3] Roy Fielding, Jim Gettys, Jeffrey Mogul, Henrik Frystyk, Larry Masinter, Paul Leach, and Tim Berners-Lee. Hypertext transfer protocol–http/1.1, 1999.

[4] Peter Saint-Andre et al. Rfc 3920: Extensible messaging and presence protocol (xmpp): Core. *Internet Engineering Task Force*, 2004.

[5] Andrew Banks and Rahul Gupta. Mqtt version 3.1. 1. *OASIS standard*, 29:89, 2014.

[6] John O'Hara. Toward a commodity enterprise middleware. *Queue*, 5(4):48–55, May 2007.

[7] Z. Shelby, A. P. Castellani, and C. Bormann. Coap: An application protocol for billions of tiny internet nodes. *IEEE Internet Computing*, 16(02):62–67, mar 2012.

[8] T. Yokotani and Y. Sasaki. Comparison with http and mqtt on required network resources for iot. In *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, pages 1–6, Sep. 2016.

[9] N. Naik. Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http. In *2017 IEEE International Systems Engineering Symposium (ISSE)*, pages 1–7, Oct 2017.

[10] Gaston C Hillar. *MQTT Essentials-A Lightweight IoT Protocol*. Packt Publishing Ltd, Birmingham, UK, 2017.

[11] D. Dragomir, L. Gheorghe, S. Costea, and A. Radovici. A survey on secure communication protocols for iot systems. In *2016 International Workshop on Secure Internet of Things (SIoT)*, pages 47–62, Sep. 2016.

[12] Hyun Cheon Hwang, JiSu Park, and Jin Gon Shon. Design and implementation of a reliable message transmission system based on mqtt protocol in iot. *Wireless Personal Communications*, 91(4):1765–1777, Dec 2016.

[13] S. Shin, K. Kobara, Chia-Chuan Chuang, and Weicheng Huang. A security framework for mqtt. In *2016 IEEE Conference on Communications and Network Security (CNS)*, pages 432–436, Oct 2016.

[14] K. Fysarakis, I. Askoxylakis, O. Soultatos, I. Papaefstathiou, C. Manifavas, and V. Katos. Which iot protocol? com-paring standardized approaches over a common m2m application. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7, Dec 2016.

[15] M. Singh, M. A. Rajan, V. L. Shivraj, and P. Balamuralidhar. Secure mqtt for internet of things (iot). In *2015 Fifth International Conference on Communication Systems and Network Technologies*, pages 746–751, April 2015.

[16] S. Andy, B. Rahardjo, and B. Hanindhito. Attack scenarios and security analysis of mqtt communication protocol in iot system. In *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pages 1–6, Sep. 2017.

[17] Krešimir Grgić, Ivan Špeh, and Ivan Heđi. A web-based iot solution for monitoring data using mqtt protocol. In *2016 International Conference on Smart Systems and Technologies (SST)*, pages 249–253. IEEE, 2016.

[18] Aimaschana Niruntasukrat, Chavee Issariyapat, Panita Pongpaibool, Koonlachat Meesublak, Pramrudee Aiumsupucgul, and Anun Panya. Authorization mechanism for mqtt-based internet of things. In *2016 IEEE International Conference on Communications Workshops (ICC)*, pages 290–295. IEEE, 2016.