IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 23, Issue 2, Ser. I (Mar. – Apr. 2021), PP 01-07
www.iosrjournals.org

Modeling of a Communication protocol for IoT based Applications

Mohammad Nurus Salam
Department of Computer Science and Technology
Military Institute of Science and Technology

Abstract - *Internet of Things (IoT) connected devices will be reaching people seamlessly in future days. With the passage of time in the context of IoT, many protocols have been devised for secured transmission, including XMPP, CoAP AMQP, ZigBee, 6LoWPAN, LWM2M etc. In Addition to above, The Message Queue Telemetry Transport (MQTT) is extensively used protocols in most IoT based communication network. The security aspects for the IoT domain is an open field of research and analysis. Since, MQTT standard has no mandatory requirements regarding the security services; therefore, manipulating the security issues in MQTT platforms seems very easy. This paper looks to evaluate the proposed security enhanced IoT protocols with the security analysis with a focus to MQTT protocol. Basing on test result and analysis, a Secured Message Queue Telemetry Transport (SMQTT) protocol is proposed. The protocol is based on cryptographic primitives to offer security services for this IoT system. In doing that, a formal verification for a SMQTT protocol is conducted by Prover if to prove that the proposed protocol satisfies the intended security attributes. This verification will cover most of testing scenarios that may have been ignored in the original protocol standard. The evaluation metrics includes: confidentiality protection, integrity protection, authentication mechanism etc.*

Key words: IoT, MQTT, SMQTT, 6LoPAN, CoAP, ZigBee, Authentication, Encryption, Cryptography, ProVerif

---

Date of Submission: 25-02-2021 Date of Acceptance: 10-03-2021

I. Introduction

Number of IoT connected devices worldwide is likely to be increased to 43 billion by 2023, a threefold increase from 2018 [1]. IoT-based application has diverse usage, for example, in automated fire control, logistics and energy management, smart health monitoring system, robotics, military surveillance and weapon system and so on [2]. IoT-based systems are equipped with wireless functionality along with sensors, communication channel between devices and back-end systems. Security aspects of IoT world are still an open field of research. Most of the IoT protocols work in network layer,

IoT layer	Technologies
Sensing layer	Sensor-networks, RFID, cameras, radars etc.
Network layer	ZigBee, Bluetooth, Wi-Fi, 6LoPAN, mobile networks, GPS etc.
Application layer	smart home, energy/power management, selfdriving cars, cloud technology etc

Table 1: IoT three-layer stack

IoT-based applications demonstrated mentionable security vulnerabilities. Few of the such surfaces are: IoT devices (i.e. sensors and actuators), IoT-specific applications, backend data storage and most importantly communication channels between the devices as well as between the devices the back-end system, etc. IoT based platforms use several communication protocols; i.e. CoAP, AMQP, MQTT and many other. Due to constraint environment in IoT, most of the protocols does not provide complete information security services. MQTT is one of the widely used IoT communication protocols; MQTT standard has no specific requirements about the security standards. IoT developers use this protocol because of its minimal bandwidth requirement and low memory consumption [3]. Sometimes, IoT device sends confidential data that should only be accessed by authorized people or devices. The MQTT protocol only provides authentication for the security mechanism and it does not encrypt the data in transit. Thus, data privacy, authentication, and data integrity become a concern in MQTT implementation. As such, after finding the specific vulnerabilities of MQTT protocol, while proposing a secured MQTT protocol, verification by a standard verifier, i.e ProVerif, would be effective. Identified vulnerabilities shall be analyzed and removed in proposing a security enhanced similar protocol to be deployed for running IoT-based applications. The rest of the paper is organized as follows: Section II introduces the

MQTT protocols in broader perspective along with short highlights on the security aspects of IoT protocols, Section III discusses about the related studies and works in recent past. An Overview about Prover if Cryptographic Tools is discussed in Section IV. In Section V Formal Modeling for the Proposed Communication Protocols using ProVerif is presented. Finally, the paper is concluded in Section VI.

II. IoT Protocols

This section firstly introduces the background theories followed by the specific understanding on the security of MQTT protocol. At the end of this section, a critical summary is presented to highlight the research gap and motivation to this research.

A. Introduction to IoT Protocols

IoT is about interconnecting a system, uniting together two emerging technologies: wireless connectivity and sensors. These connected embedded systems are independent microcontroller-based computers that use sensors to collect data from a network. This concept emerged a long time ago, through mentionable development in sensing technology and objects connected to the internet. With current internet infrastructure, wireless communication plays a vital role in IoT devices allowing them to transmit messages. Therefore, the vitality of these messages lies in authentication and security. Numerous key management techniques have also been introduced to provide a secured transmission over the internet.

The Internet of Things (IoT) is composed of physical objects embedded with electronics, software, and sensors, which allows objects to be sensed and controlled remotely across the existing network infrastructure. It facilitates direct integration between the physical world and computer communication networks. It significantly contributes to enhanced efficiency, accuracy, and economic benefits. Therefore, IoT has been widely applied in various applications such as environment monitoring, energy management, medical healthcare systems, building automation, and transportation.

Unfortunately, due to the security constraints of IoT domain and related privacy of IoT users, it exposes a versatile threat to the whole system. Therefore, design of enhanced IoT based secured protocol is a crucial issue. Although the recent goal in the IoT industry is on the ease of use, to improve functional properties, and optimize costs, simultaneously there is an urgent need to evaluate its of security standards of IoT protocols.

With current internet infrastructure, wireless communication plays a vital role in IoT devices allowing them to transmit messages. Therefore, the vitality of these messages lies in authentication. Numerous security techniques have also been introduced to provide a secured transmission over the internet.

B. Salient Aspects of Different Protocols

However, all complete IoT systems are the same in that they represent the integration of four distinct components: sensors/devices, connectivity, data processing, and a user interface. Every IoT protocol has its own merits and demerits. ZigBee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection. Constrained Application Protocol (CoAP) is a specialized Internet Application Protocol for constrained devices, as defined in RFC 7252. It enables those constrained devices called "nodes" to communicate with the wider Internet using similar protocols. CoAP is designed for use between devices on the same constrained network (e.g., low-power, lossy networks), between devices and general nodes on the Internet, and between devices on different constrained networks both joined by an internet. CoAP is also being used via other mechanisms, such as SMS on mobile communication networks. CoAP is a simplification of the HTTP protocol running on UDP, that helps save bandwidth.

C. MQTT Protocol

The Message Queue Telemetry Transport (MQTT) protocol is regarded as one of the best participant protocols for the IoT fields as it is a high time lightweight publisher and subscriber-based protocol. The MQTT is consisted of five main components, those are: The Broker: It is the worker that gets and distributes messages between customers.

a. The Message: It is the holder of the information that has been shipped off the agent by the distributer or has been gotten by the supporter from the intermediary.

b. The Publisher: It is the gadget which sends messages to the representative to refresh the information of certain topics.

c. The Subscriber: It is the gadget which gets messages from the representative that convey the refreshed status of the agent's topics.

d. The Topic: It is an element on the dealer where the distributor sends messages to it and the supporter gets messages from it.

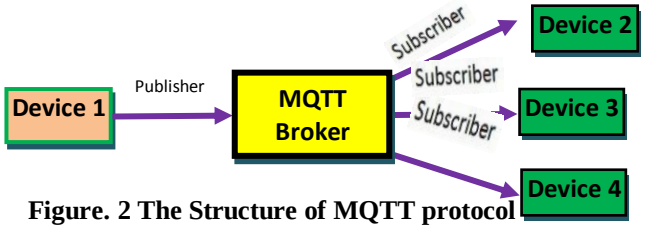

Figure. 2 The Structure of MQTT protocol

The authority of MQTT standard delivered by the Organization for the Advancement of Structured Information Standards (OASIS) doesn't have obligatory necessities with respect to the security administrations like validation, privacy, information honesty, and access control [11]. Presently, tackling the security related issues is an undertaking or potentially execution explicit issue and there is no particular normalization to deal with these issues.

D. Security Aspects of MQTT Protocols

MQTT protocol (ISO / IEC 20922: 2016) is one of the protocols that are already standardized by ISO. Many IoT developers use this protocol because of its minimal bandwidth requirement and low memory consumption. Sometimes, IoT device sends confidential data that should only be accessed by authorized people or devices. Unfortunately, the MQTT protocol only provides authentication for the security mechanism which, by default, does not encrypt the data in transit. Thus, data privacy, authentication, and data integrity become problems in MQTT implementation. This paper discusses several reasons on why there are many IoT systems that do not implement adequate security mechanism. Next, it also demonstrates and analyzes how we can attack this protocol easily using several attack scenarios. Finally, after the vulnerabilities of this protocol have been examined, we can improve our security awareness especially in MQTT protocol and then implement security mechanism in our MQTT system to prevent such attack.[4]

In order to secure MQTT there are commonly used approaches like Advanced Authentication Mechanisms, Authorization, TLS / SSL, Securing MQTT Systems etc. There are few other security concepts and implementations with MQTT: X509 Client Certificate Authentication, OAuth 2.0, Payload Encryption, Message Data Integrity etc.

Security in MQTT is divided in multiple layers. Each layer prevents different kinds of attacks. The goal of MQTT is to provide a lightweight and easy-to-use communication protocol for the Internet of Things. The protocol itself specifies only a few security mechanisms. MQTT implementations commonly use other state-of-the-art security standards: for example, SSL/TLS for transport security. Since security is difficult, it makes sense to build upon generally accepted standards [5].

III. Related Works

A number of researches have been undertaken focusing to MQTT security vulnerability and its enhancement on security aspects. This section briefly introduces the work related to the design, development and usability of the secured MQTT protocol.

A. Token Based Authentication

Bhawiyuga et al. in 2017[6] proposed a token-based authentication for MQTT utilizing a JSON Web Token (JWT) worker as a confirmation worker. They select the JWT on the grounds that it has a little message size. They proposed a framework design in which the client sends his/her username and secret word to the JWT authentication worker. At that point, the worker checks its information base for the legitimacy of the client certification. In the event that they are legitimate, the worker sends the token to the client who saves that token in his/her nearby stockpiling. When the client needs to interface with the Broker, he/she sends his/her token during the association foundation stage to the Broker who checks the legitimacy of the token with the JWT worker. In the event that it is substantial, the Broker will permit the client to distribute/buy in to the required subjects. The succession graph of their framework is demonstrated in Fig. 2

It is as per the following:

a. The customer demands a token from the validation worker utilizing its username and secret key to confirm itself.

b. The validation worker awards token to the customer after approving its Certifications.
c. The customer utilizes the token in the association establishment stage with the MQTT Broker.
d. The Broker checks the legitimacy of the token introduced by Customer with the confirmation worker.
e. The confirmation worker answers with the legitimacy status of the token to the MQTT Broker.
f. If there should be an occurrence of legitimate token, the Broker affirms the association demand from the customer.
g. The customer begins to get to the subjects of the Broker.

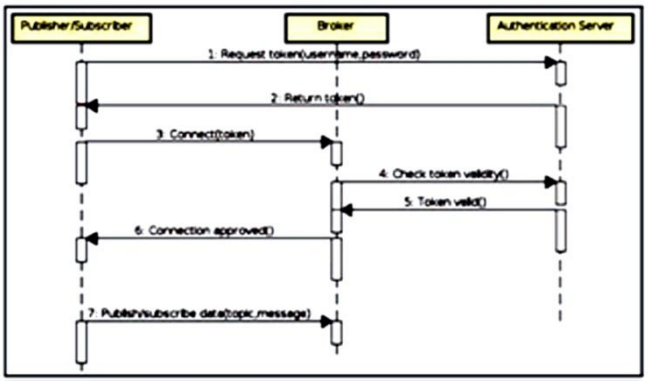

Figure 3: The Sequence Diagram of the token Based MQTT Publisher/Subscriber System

B. OAuth 1.0a Approval Standard

Niruntasukrat et al. [7] introduced an approval instrument for MQTT utilizing OAuth 1.0a approval standard. They stated that since OAuth 2.0 [12] doesn't uphold any security on top of the TLS/SSL, OAuth 1.0a is more reasonable for the IoT climate than OAuth 2.0. Their thought can be summed up in that the client who has the entrance accreditations will designate a portion of his/her position to certain gadgets. Their proposed component is introduced in Fig. 3. It has the accompanying advances:
a. The client sends a HTTPS message to the AuthServer to demand the Device ID and its mystery.
b. The AuthServer awards the gadget qualifications to the client (Gadget ID and its mystery).
c. The client physically implants the gadget qualifications into the gadget nearby memory.
d. The gadget ships off the AuthServer to demand a Request Token. This message is carefully marked utilizing the HMAC- SHA1 calculation where the HKey is the Device Secret.
e. The AuthServer issues a Request Token and its mystery after approving the gadget qualifications.
f. The gadget ships off the AuthServer to demand an Access Token. This message is carefully marked utilizing the HMAC- SHA1 calculation where the HKey is the Request Token Secret what's more, the Device Secret.
g. The AuthServer demands client endorsement utilizing email or Short Message Service (SMS).
h. The client supports the Device ID and the entrance advantage scope.
i. The AuthServer awards the Access Token and its mystery to the gadget.
j. The gadget can get to the MQTT dealer where the username will be the Device ID with connected timestamp and the secret word will be created from the entrance token and be the Access Token Secret and the Device Secret.

C. Attribute Based Encryption (KP/CP ABE)

Rahman et al. in 2018 [8] offered the use of Key Policy/ Code Policy Attribute Based Encryption (KP/CP ABE) utilizing Elli spasm Curve Cryptography (ECC) to get a changed MQTT convention fit for conveying secure correspondence between end gadgets. The arrangement chart of their proposed framework architecture is appeared in Fig. 4. This design has the accompanying stages:
a. After framework introduction, both the Device and the Web Workers will enlist in the MQTT Broker.
b. The key administration stage is performed among the MQTT Broker, the IoT Device and the Web Server.
c. Both the Device and the Web Server will buy in the required subjects of the MQTT Broker.
d. At the point when an approved customer sends an order to the Web Worker, it will encode this order and distribute the encoded message to the MQTT Broker.
e. The Broker will pass the encoded message to the Device where the decoding cycle will be performed and the fitting action(s) will be taken.
f. The gadget will encode the readied reaction and distribute the encoded message to the MQTT Broker.
g. The MQTT Broker will pass the encoded reaction to the Web Server.
h. The Web Server will decode the reaction.
i. The got decoded reaction is conveyed to the customer.

D. Use of Lightweight Cryptography

Bali et al. in 2019[9] handled a lightweight instrument for confirmation in MQTT stages utilizing riotous calculation with block cypher. Their introduced a simulation model as shown in Fig. 5. They referenced that their proposed approach relies upon the high variety of the turbulent calculation and complex algorithm. Additionally, they expressed that as the variety of the keys will incur trouble since it will be a difficult task to get back the plaintext. Hence, a safer framework is accomplished. Also, they explained that they kept up the high variety between the back-to-back keys by appropriately choosing the turbulent boundaries and they relies upon the distance entropy while choosing these MQTT based boundaries.

E. Summary of the Studies

After conducting thorough analysis of the mentioned studies, it can be identified that each of them fulfills one or few aspects of the security requirement like authentication, confidentiality, data integrity, authorization etc.

IoT technology offers huge opportunities and also brings many new challenges related to the authentication in IoT devices. Using passwords or pre-defined keys have drawbacks that limit their use for different IoT applications. Thus, authenticating users basing on password mechanism not only meet the purpose. As such, Token-Based Lightweight User Authentication (TBLUA) for IoT devices, which is based on token technique in order to enhance the robustness of authentication, is commonly used in present days. Tokens work like a stamped ticket. The user retains access as long as the token remains valid. Once the user logs out or quits an app, the token is invalidated. Tokens offer a second layer of security, and administrators have detailed control over each action and transaction.

Many social media platforms use the OAuth 1.0a method to act, or make API requests. But there are common errors like access token failure, matching of token, token expired, refusal of timestamp etc. There are other issues like signing every request, addressing native applications and separation of roles. This is eliminated in OAuth2.0 where multiple flows are presented deliberately.

Attribute-based encryption is a type of public-key encryption in which the secret key of a user and the ciphertext are dependent upon attributes (e.g. the country in which they live, or the kind of subscription they have). The systems suffer mainly from two drawbacks: non-efficiency and non-existence of attribute revocation mechanism. Other few challenges are: Key coordination, Key escrow, Key revocation etc. The motivation of lightweight cryptography is to use less memory, less computing resource and less power supply to provide security solution that can work over resource-limited devices. The lightweight cryptography is expected simpler and faster compared to conventional cryptography.

However, since the lightweight cryptographic algorithms are designed to handle small amounts of information, they do not have high bandwidth. The very existence of constraints says that light ciphers primarily designed not to soft but to hardware implementation. The inherent disadvantage of lightweight cryptography is less secured.[7]-[10]

IV. Use of ProVerif for IoT based Security Testing

ProVerif is an well-organized computerized tool used during the verification testing stage of the any security protocol. It is based on Pi calculus and it has the ability to verify the authenticity and the secrecy properties of the cryptographic security protocols. It can handle an limitless number of sessions for the protocol. It can also monitor the communication where it can capture, adapt, insert, and regenerate messages to spitefully attack the system. Besides, ProVerif provides a tracing for the adversary attack to the system to clarify whether the protocol has security problems or not.

This tool verifies the protocol for an unbounded number of runs (sessions), using unbounded message space. It has been used and developed since 2001. ProVerif has been successfully used to automatically analyse the security of cryptographic protocols used in electronic voting or key exchange. In this paper, we introduced a short usage of Proverif, the tool for the verification of the security protocol. We also showed how the tool works internally when verifying a protocol, ProVerif seems to be fairly accessible to all kind of users because of its user friendliness and ease of understanding.[11]

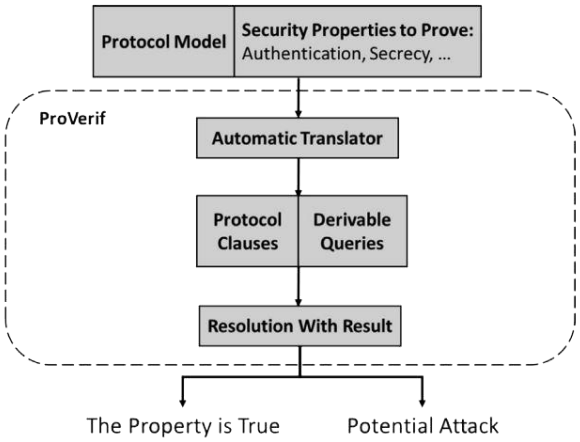

Figure 4: ProVerif Testing Sequence

V. Formal Modeling of the Protocol

The formal verification of the SMQTT protocol is completed by using ProVerif version 2.00. It is done in two stages:

a. The first stage is the validation by Broker to identify a PC or a Dr using Elliptic Curve Digital Signature Algorithm algorithm (ECDSA).

The second stage is the validation of:

b. The secrecy of the session key generated by the PC and the Dr using the Elliptic-curve DiffieHellman (ECDH) algorithm. After that, secrecy of the message is encrypted by generating session key using a symmetric encryption algorithm.[11]

A. The authentication verification stage

The main process of the authentication formal verification stage is shown in Figs. 13–15. Two different processes are required to strictly check the security aspects of the client validation in the SMQTT protocol. Those are: the client process and the Broker process. The client public key is created and is used as input to unlimited number of the Client and Broker process. In the client process and the Broker process the messages of the ECDSA is built and exchanged between the two parties using the constructors.

The authentication query is checked to verify that the end Authentication Check event is reached securely without any attack possibility after the begin Authentication Check. The ProVerif results to verify this query. Tracing the results, one can conclude that the tested authentication query is true.

B. The verification stage of the session key and the encrypted message secrecy

Three different processes are needed to formally check the secrecy of the generated session keys and the secrecy of the encrypted message in the SMQTT protocol - the PC, the Dr and the Broker process.

In the main process limitless number of the PC process is created in parallel with infinite number of the Dr process along with number of the Broker process. During the PC process, the Dr process and the Broker process the messages of the ECDH is built and swapped between the PC and the Dr through the Broker using the declared constructors. Therefore, the session key is generated between the PC and the Dr. After that, the message is evenly encrypted using the generated session key.

The session key secrecy request is confirmed to validate whether the created session key is kept secret between the PC and the Broker or an opponent can expose it. Moreover, the message secrecy query is verified too to confirm that only the Dr is the one who can go through the messages sent by the PC and vice versa. The ProVerif results to authenticate those queries. By conducting observant study of the results, one can infer that both the message's secrecy and the session key secrecy queries are verified by Proverif as true or not. So, the channel between the Dr and the PC through the Broker is reliable from a security consideration and they can sucre exchange of data is possible.

C. The security objectives of the SMQTT protocol

After carrying out laborious analysis for the security queries using automated verifier tool ProVerif, it is found that the following security objectives are achievable by the SMQTT protocol:

a. Successful Verification of the Client: The client identification is successfully authenticated by the Broker. This is cleared by the correctness of the following query: inj-event (endAuthentication Check(id)) ==> inj-event (begi nAuthentication Check(id))

b. Secrecy of the the Session Key: The value of the session key is only known to the PC and the Dr. This is proven by the attacker failure to resolve the session key as indicated by the f query: query attacker (session_key)

c. Secrecy of the Message: The content of the can only be read by the PC and the Dr. This is depicted by the opponent failure to reveal the contents of the swapped messages between the PC and the Dr as displayed by the query: query attacker (msg)

VI. Conclusion

In today's world, cyber security and IoT security don't go together. There are no industry standards for architecture or IoT security. Devices often use custom-built operating systems and proprietary communication protocols. IoT security continues to remain as a true obstacle. Thus it is always so hard to secure IoT devices and protocols. MQTT is one of the widely used protocols used in IoT system where there are significant vulnerabilities. Security aspects of MQTT protocols have been studied form different perspective. Yet, the optimum result is not attained. In this paper, considering the threats and vulnerabilities of the protocol, a secured MQTT protocol is proposed which is further verified by a smart cryptographic tool, Proverif.

The proposed SMQTT protocol is deliberately tested by ProVerif and the security of the networks between the PCs and the Drs across the Broker are preserved. Thus, the designed SMQTT protocol can be applied over an untrusted network proposing secure communications between the PCs and their Drs. If the Broker exists in the cloud having a great effect on the cost reduction while applying the proposed SMQTT protocol on a real time traditional MQTT based IoT network.[9]-[12]

References:

[1]. S. R. J. Ramson, S. Vishnu, and M. Shanmugam, "Applications of Internet of Things (IoT) - An Overview," in 2020 5th International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, Mar. 2020, pp. 92–95, doi: 10.1109/ICDCS48716.2020.243556.

[2]. V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," IEEE Access, vol. 7, pp. 82721–82743, 2019, doi: 10.1109/ACCESS.2019.2924045.

[3]. TONIN, M.: The Internet of Things: Promises and Perils of a Disruptive Technology, NATO Report, 2017. https://www.nato-pa.int/document/2017-internet-things-toninreport-175-stctts-17-e-bis (letöltve: 2018. 04. 29.)

[4]. MQTT IoT Protocol complete Tutorial - How it Works with a demo, 1Sheeld | All Arduino shields on your Smartphone, 04-Jul-2018

[5]. M. Singh, M. A. Rajan, V. L. Shivraj and P. Balamuralidhar, "Secure MQTT for Internet of Things (IoT)," 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, 2015, pp. 746-751.

[6]. Bali RS, Jaafar F, Zavarasky P. Lightweight authentication for MQTT to improve the security of IoT communication. In: Proceedings of the 3rd International Conference on Cryptography, Security and Privacy - ICCSP '19, Kuala Lumpur, Malaysia; 2019. p. 6–12.

[7]. E. Elemam, A. M. Bahaa-Eldin, N. H. Shaker, and M. Sobh, "Formal verification for a PMQTT protocol," Egyptian Informatics Journal, vol. 21, no. 3, pp. 169–182, Sep. 2020, doi: 10.1016/j.eij.2020.01.001.

[8]. [8] Riahi Sfar A, Natalizio E, Challal Y, Chtourou Z. A roadmap for security challenges in the Internet of Things. Digit Commun Netw, Apr 2018;4 (2):118–37.

[9]. Niruntasukrat A, Issariyapat C, Pongpaibool P, Meesublak K, Aiumsupucgul P, Panya A. Authorization mechanism for MQTT-based Internet of Things. In: 2016 IEEE International Conference on Communications Workshops (ICC), Kuala Lumpur, Malaysia; 2016, pp. 290–295.

[10]. Blanchet B, Smyth B, Cheval V, Sylvestre M. ProVerif 2.00: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial. Fig. 20. The Results for the Formal Verification of the Session Key Secrecy and the Encrypted Message Secrecy. E. Elemam et al. / Egyptian Informatics Journal 21 (2020) 169–182 181

[11]. Bali RS, Jaafar F, Zavarasky P. Lightweight authentication for MQTT to improve the security of IoT communication. In: Proceedings of the 3rd International Conference on Cryptography, Security and Privacy – ICCSP '19, Kuala Lumpur, Malaysia; 2019. p. 6–12.

[12]. Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA). [Online]. Available: https://tools. ietf.org/html/rfc6979. [Accessed: 27-Aug-2019].

Mohammad Nurus Salam. "Modeling of a Communication protocol for IoT based Applications." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 23(2), 2021, pp. 01-07.