Evaluation of the impact on energy consumption of MQTT protocol over TLS

Edgaras Baranauskas
Department of Computer Sciences
Kaunas University of Technology
Kaunas, Lithuania
edgaras.baranauskas@ktu.edu

Jevgenijus Toldinas
Department of Computer Sciences
Kaunas University of Technology
Kaunas, Lithuania
eugenijus.toldinas@ktu.lt

Borisas Lozinskis
Department of Computer Sciences
Kaunas University of Technology
Kaunas, Lithuania
borisas.lozinskis@ktu.lt

***Abstract*— Message Queuing Telemetry Transport (MQTT) protocol is widely used in device-to-device communications. While MQTT has three quality of service (QoS) levels, it does not integrate security mechanisms. Transport Layer Security (TLS) is the standard protocol on top of the Transmission Control Protocol (TCP) to secure data in communications. In this paper, we evaluate the impact on energy consumption of MQTT protocol using its QoS levels over TLS.**

Keywords—IoT, MQTT, TLS, battery energy consumption

I. INTRODUCTION

According to Gartner prediction spending on Internet of Things (IoT) endpoint security solutions worldwide will reach 631 millions of dollars in 2021 [1]. The term Internet of Things define smart objects that are interconnected using various network interfaces and protocols such as Constrained Application Protocol (CoAP), Message Queuing Telemetry Transport (MQTT), MQTT-SN (for sensor networks), Extensible Messaging and Presence Protocol (XMPP), Web Application Messaging Protocol (WAMP) and many others. In machine-to-machine (M2M) application layer protocols most popular is MQTT, well-known cloud platforms, such as Amazon AWS, Microsoft Azure, and IBM Watson expose their services through MQTT [2]. MQTT has a low memory footprint, low power consumption, and better distribution of information to recipients [3]. Because of that, MQTT protocol is widely used in device-to-device (D2D) communications, where one of the major issue is to ensure the security of devices and D2D communications [4]. MQTT has three quality of service (QoS) levels and does not integrate security mechanisms. Transport Layer Security (TLS) is the standard protocol on top of the Transmission Control Protocol (TCP) to secure data in communications. OASIS [5] explicitly recommends the utilization of TLS as security decision at transport layer. In this paper, we evaluate the impact on energy consumption of MQTT protocol and its QoS levels over TLS.

II. RELATED WORK

In [4] authors declare the user's responsibility to address security issues for MQTT, MQTT-SN protocols and suggests enabling security for them by envisaging SSL/TLS, but due to IoT heterogeneity it is cumbersome to manage certificates and keys. Thus, authors [4] propose attribute based encryption for secure MQTT that augments security feature for the existing MQTT protocol and its variants. Use of Datagram Transport Layer Security (DTLS) for securing data communications over User Datagram Protocol (UDP) adds at least 33 bytes to the original packet header, and while IoT devices run on batteries, efficient secure communication scheme is needed [6].

A novel security mechanism introduced for MQTT environments is based on AugPAKE via a secure side channel, where authentication and authorization tokens are transported in the same field [7] of the topic name. In [8] the most known application layer protocols are compared: CoAP, MQTT, XMPP, HTTP, AMQP and WebSocket, All the protocols mentioned above use TCP as transport layer (CoAP uses UDP) and TLS/SSL as security layer (CoAP uses DTLS). In terms of Message Oriented Approach (MOA), MQTT stands out [8]. Requirements for authentication, authorization, data integrity, and confidentiality do not included in the MQTT specification. Authors [9] argue that the lack of security requirements in the MQTT protocol standard is related to:

- MQTT focuses only on message dispatching.
- Reducing the overhead that is related to security features is used to keep the protocol as light as possible.
- Historical implementations of MQTT were based on private networks.
- Significantly different security functionalities required while MQTT is used from IoT devices to Facebook messenger mobile application.

The authors [9] are inclined to believe that a good mid-term solution to large-scale MQTT security problems could be represented by implementation of TLS. Current open-source MQTT implementations compared in table I.

TABLE I. OPEN-SOURCE MQTT IMPLEMENTATION

MQTT implementation	**MQTT implementation property**		
	Definition	***Security***	***QoS***
Mosquitto [10]	Most commonly used implementation	SSL/TLS support	QoS0, QoS1, QoS2
eMQTTC [11]	Asynchronous Erlang MQTT Client Requires Erlang R17+	TCP/SSL Socket Support	QoS0, QoS1, QoS2
Apollo [12]	Is a faster, more reliable, easier to maintain messaging broker built from the foundations of the original ActiveMQ	SSL/TLS Support	QoS0, QoS1, QoS2
Artemis [13]	Implementation arising from ActiveMQ	SSL support	QoS0, QoS1, QoS2

In [14] proposed potential methodologies to extend the Common Architectures and Network services found in the IEEE 1451 Family of Standard into applications that utilize MQTT. The authors installed the Mosquitto MQTT client

onto ESP-32s, MQTT broker onto Raspberry Pi 3 and experimentally conclude that MQTT is an effective communication protocol when it comes to small-scale systems, security is a major area for future investigation. MQTT has its downsides in security but is being greatly adapted in the world of IoT today and the hope is extend that adaptation to the IEEE 1451 Family of standards [14].

MQTT is a simple protocol designed for devices with low processing power and it tries to minimize the processing needed to exchange messages, which means that serious security problems arise such as lack of: authentication, authorization, confidentiality and integrity [15].

The security challenges of the IoT industry with focus on standardized communication protocols explored and implementation details for the security levels mandated by the Constrained Application Protocol provided in [16]. MQTT implementations also offer out of the box the security certificates mode that could be achieved in the Java Paho library or as part of the Mosquitto framework. In fact, the MQTT broker also offers the possibility to maintain a list of revoked certificates that can be used to disable rogue endpoints [16].

The most critical issues with the aim of guiding future research directions on the IoT security panorama highlighted [17]. According to the author conclusion, the most vulnerable level of the IoT system model is the perception layer due to the physical exposure of IoT devices, to their constrained resources and to their technological heterogeneity. Thus, it is crucial, in the next future, to start working on the critical issues of this level implementing lightweight security solutions that can adapt to the heterogeneous environments with resource-constrained devices.

Smart city solutions have to be energy-efficient, cost-efficient, reliable, secure, to do that IoT devices should operate in a self-sufficient way without compromising QoS in order to enhance the performance with uninterrupted network operations. Therefore, the energy efficiency and life span of IoT devices are key to next generation smart city solutions [18]. With the increase in IoT applications for smart cities, energy-efficient solutions are also evolving for low-power devices. Energy-efficient solutions such as Lightweight Protocols, Scheduling Optimization, and Predictive Models for Energy Consumption, Cloud-Based Approach, Low-Power Transceivers, and Cognitive Management Framework can reduce energy consumption or optimize resource utilization. Possible future directions for energy management in smart cities are [18]:

- Energy-efficient mechanisms for software-defined IoT solutions, which can provide scalable and context-aware data and services.
- Directional energy transmission from dedicated energy sources for wireless power transfer.
- Energy efficiency and complexity of security protocols are crucial aspects for their practical implementation in IoT; thus, it is important to investigate robust security protocols for energy constraint IoT devices.
- Fog computing can lead to energy saving for most of the IoT applications; therefore, it is important to study energy consumption of fog devices for IoT applications.

In [19] authors evaluate MQTT (QoS0) vs HTTPS, send performance, battery energy consumption and conclude that while HTTPS is slightly more efficient in terms of establishing connection, MQTT is much more efficient during transmission.

III. MQTT QUALITY OF SERVICE LEVELS

MQTT provides three levels of QoS [20]:

At most once (Fig. 1) - sometimes called "fire and forget". The message is delivered at most once, or it is not delivered at all.

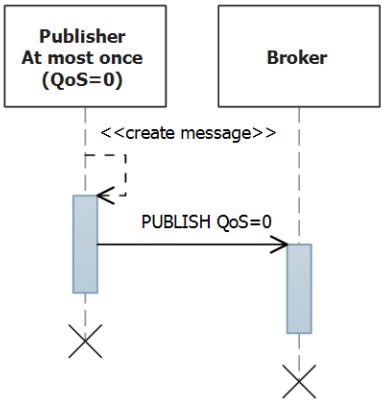

Fig. 1. MQTT QoS level “At most once"

At least once (Fig. 2), it is the default mode of transfer. The message is always delivered at least once. If the sender does not receive an acknowledgment, the message is sent again with the DUP flag set until an acknowledgment is received.

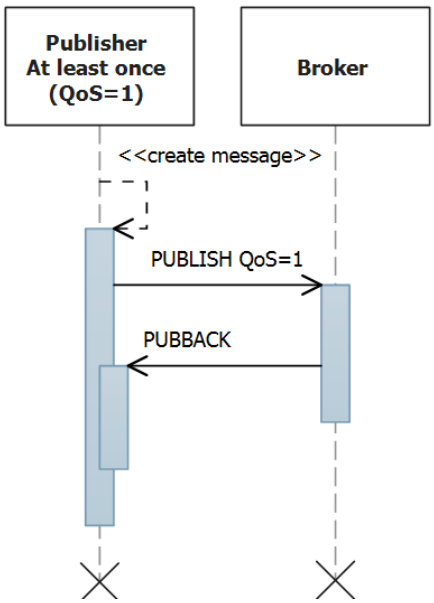

Fig. 2. MQTT QoS level “At least once"

Exactly once (Fig. 3), the message is always delivered exactly once. The message must be stored locally at the sender and the receiver until it is processed. **Exactly once** is the safest, but slowest mode of transfer.

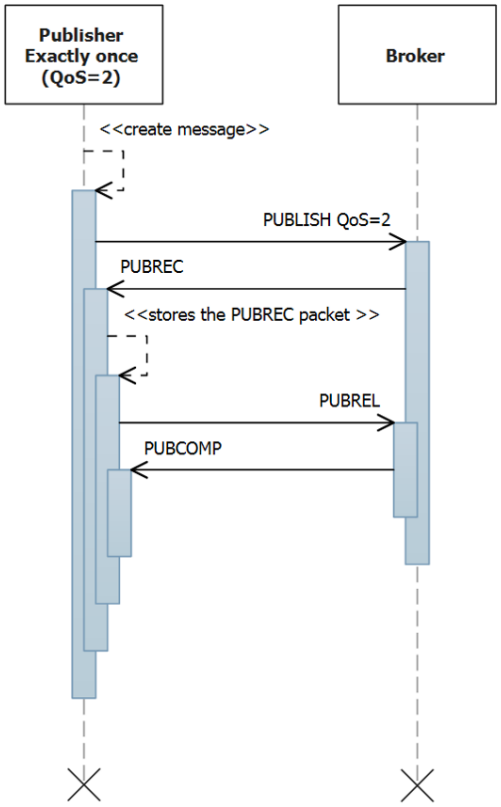

Fig. 3. MQTT QoS level “Exactly once "

IV. Evaluation Framework and Experimental Setup

A general framework for evaluation of the impact on energy consumption of MQTT protocol over TLS is shown in Fig. 4. T-diagram is linking together three evaluation domains: security, reliability and energy consumption.

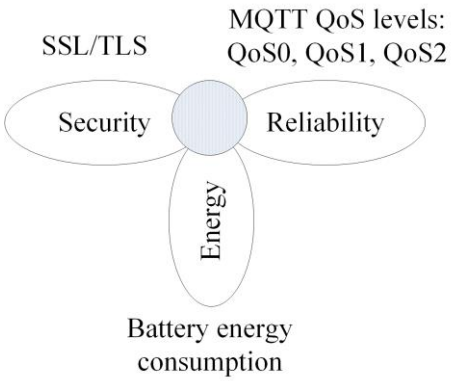

Fig. 4. General framework for evaluation of the impact on energy consumption of MQTT protocol over TLS

The framework also outlines a context of the selected domains: for security domain it is SSL/TLS , the MQTT QoS levels ensure reliability, and battery energy consumption measurement for energy domain. Our experiments are performed using (see Fig. 5):

- **Access point** – Wi-Fi router TP-Link.
- **Broker** – Raspberry Pi2 with Broadcom BCM2837 Arm7 Quad Core CPU, clock frequency 900MHz, 1GB RAM, 802.11b/g/n Wi-Fi communication protocols.
- **Subscriber/Publisher** – IoT Module ESP32 with Tensilica L106, 32-bit, RISC CPU, clock frequency 160 MHz, 802.11b/g/n Wi-Fi communication protocols.
- **Measuring instrument** – digital multimeter MASTECH MS8050.
- **Power supply** for ESP32 – llithium battery LS903052, 3.7V, 1200mAh.

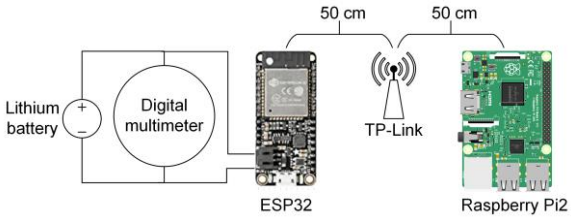

Fig. 5. Experimental setup

The ESP32 module integrates ESP8266EX is and is recommended for tests or for further development. For our evaluation, we use the Mosquitto MQTT broker that configured to use TLS. We create a simple scenario to establish encrypted connection between broker and client similarly as encrypted connection between web server and web client. To create certificates we use OpenSSL v1.1.1a software for Windows [21]. In our case, we create Certification authority (CA) in a computer with Windows OS. Certificate creation and installation in the Mosquitto MQTT broker (in our case Raspberry Pi2) and in the subscriber/publisher (in our case ESP32) is shown in Fig.6.

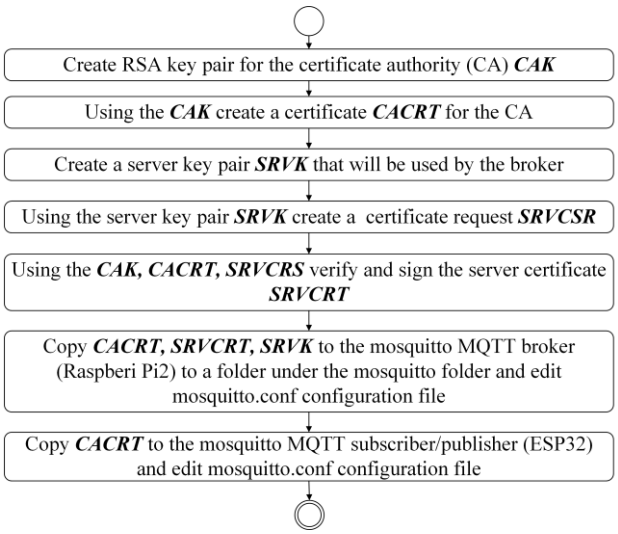

Fig. 6. Certificate creation and installation in the mosquitto MQTT broker and subscriber/publisher

V. EXPERIMENTAL RESULTS

The results of measurements are presented in Figs. 7-9. Fig. 7 shows the battery voltage level for MQTT "At most once" over TLS,

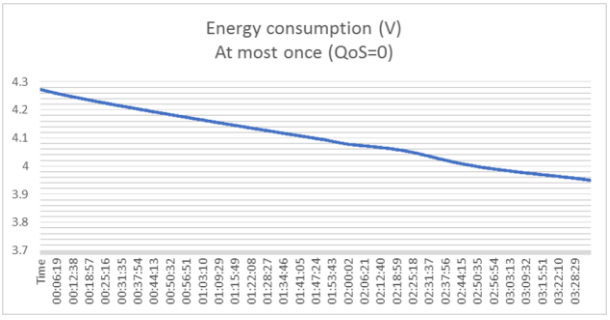

Fig. 7. Battery voltage level for MQTT "At most once" over TLS

Fig. 8 shows the battery voltage level for MQTT "At least once" over TLS.

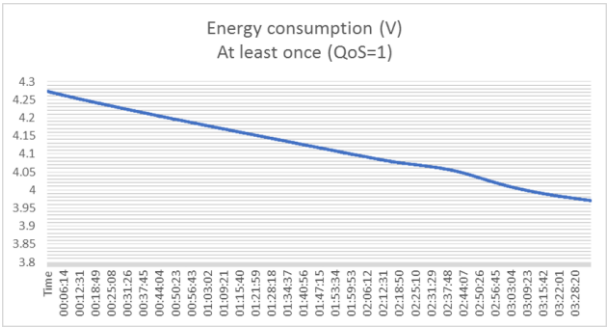

Fig. 8. Battery voltage level for MQTT "At least once" over TLS

Fig. 9 shows the battery voltage level for MQTT "Exactly once" over TLS.

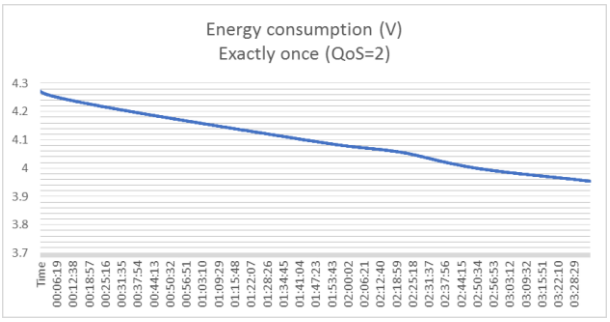

Fig. 9. Battery voltage level for MQTT "Exactly once" over TLS

The results of measurements are summarized in table II. Based on these results we can evaluate the difference in energy consumption of three MQTT protocol QoS levels over TLS. Less energy consumes "At least once (QoS1) over TLS – voltage drop 0.3026V. Most energy consumes "At most once (QoS0) over TLS – voltage drop 0.3228V and "Exactly once (QoS2)" over TLS consumes more energy than QoS1 and less than QoS0 - voltage drop 0.3176V.

TABLE II. EVALUATION OF THE IMPACT ON ENERGY CONSUMPTION OF MQTT PROTOCOL OVER TLS

MQTT QoS Level	**Energy consumption**	
	Voltage drop (V)	***Consumed time (hh:mm:ss)***
MQTT "At most once (QoS0)" over TLS	0.3228	03:34:45
MQTT "At least once (QoS1)" over TLS	0.3026	03:34:36
MQTT "Exactly once (QoS2)" over TLS	0.3176	03:34:45

VI. CONCLUSION

The energy consumption of MQTT protocol with various QoS over TLS levels is highly different. The main results of this paper are as follows:

1) The real time measured values for energy consumption securing MQTT over TLS are achieved with various QoS levels.

2) The results of energy consumption measurements when performing secure communication using MQTT protocol over TLS can be used to reliably predict energy consumption of three QoS levels:

- QoS "At least once (QoS1)" over TLS consumes less energy than the others two QoS levels (QoS0 over TLS and QoS2 over TLS),
- QoS "At most once (QoS0)" over TLS consumes more energy than the others two QoS levels (QoS1 over TLS and QoS2 over TLS),
- QoS "Exactly once (QoS2)" over TLS consumes 5 % more energy than QoS "At least once (QoS=1)" over TLS",
- QoS " At most once (QoS0)" over TLS consumes 6,7 % more energy than QoS "At least once (QoS1)" over TLS,
- QoS "Exactly once (QoS=2)" over TLS consumes 1,7 % less energy than QoS "At most once (QoS0) over TLS".

REFERENCES

[1] Gartner Says Worldwide IoT Security Spending Will Reach $1.5 Billion in 2018. STAMFORD, Conn., March 21, 2018. Gartner, Inc. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2018-03-21-gartner-says-worldwide-iot-security-spending-will-reach-1-point-5-billion-in-2018 [Accessed February 07, 2019]

[2] R. Giambona, A. E.C. Redondi, M. Cesana, "Demonstrating MQTT+: An Advanced Broker for Data Filtering, Processing and Aggregation", In 21st ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM '18), October 28-November 2, 2018, Montreal, QC, Canada. ACM, New York, NY, USA. [Online]. Available: https://doi.org/10.1145/3242102.3243317

[3] S. B. Kenitar, S. Marouane, A. Mounir, "Evaluation of the MQTT Protocol Latency over Different Gateways", SCA2018, October 2018, Tetuan, Morocco. [Online]. Available: https://doi.org/10.1145/3286606.3286864

[4] M. Singh, R. MA, S. VL, and B. P, "Secure MQTT for Internet of Things (IoT)".2015 Fifth International Conference on Communication Systems and Network Technologies. IEEE Computer Society, 2015, pp. 746-751. DOI 10.1109/CSNT.2015.16

[5] MQTT Version 3.1.1 Plus Errata 01. OASIS Standard Incorporating Approved Errata 01. [Online]. Available: http://docs.oasis-

open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os-complete.html [Accessed February 08, 2019]

[6] M. H. Amaran, M. S. Rohmad, L. H. Adnan, N. N. Mohamed, H. Hashim, "Lightweight Security for MQTT-SN". International Journal of Engineering & Technology, 7 (4.11) (2018) pp. 223-226

[7] M. Calabretta, R. Pecori, L. Veltri, "A Token-based Protocol for Securing MQTT Communications". 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2018 . DOI:10.23919/SOFTCOM.2018.8555834

[8] A. L. Marra, F. Martinelli, P. Mori, A. Rizos, and A. Saracino, "Introducing Usage Control in MQTT". S. K. Katsikas et al. (Eds.): CyberICPS 2017/SECPRE 2017, LNCS 10683, Springer International Publishing AG 2018, pp. 35–43. https://doi.org/10.1007/978-3-319-72817-9_3

[9] G. Perrone, M. Vecchio, R. Pecori, and R. Giaffreda, "The Day After Mirai: A Survey on MQTT Security Solutions After the Largest Cyber-attack Carried Out through an Army of IoT Devices". In Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017), p.p. 246-253. DOI: 10.5220/0006287302460253

[10] Eclipse Mosquitto. [Online]. Available: http://mosquitto.org/ [Accessed February 08, 2019]

[11] Erlang MQTT Client. [Online]. Available: https://github.com/emqtt/emqttc [Accessed February 08, 2019]

[12] Apollo. [Online]. Available: http://activemq.apache.org/apollo/ [Accessed February 08, 2019]

[13] Apache ActiveMQ Artemis. [Online]. Available: http://activemq.apache.org/artemis/ [Accessed February 08, 2019]

[14] J. Velez, R.Trafford, M. Pierce, B. Thomson, E. Jastrzebski, and B. Lau, "IEEE 1451-1-6: Providing Common Network Services over MQTT". IEEE Sensors Applications Symposium (SAS). 2018. DOI: 10.1109/SAS.2018.8336750

[15] S. H. Ramos , M. T. Villalba, and R. Lacuesta, "MQTT Security: A Novel Fuzzing Approach". Wireless Communications and Mobile Computing. Volume 2018, Hindawi, 11 pages. https://doi.org/10.1155/2018/8261746

[16] S. Zamfir, T. Balan, I. Iliescu, and F. Sandu, "A Security Analysis on Standard IoT Protocols". 2016 International Conference on Applied and Theoretical Electricity (ICATE). DOI: 10.1109/ICATE.2016.7754665

[17] M. Frustaci, P. Pace , G. Aloi, and G. Fortino, "Evaluating Critical Security Issues of the IoT World: Present and Future Challenges". IEEE Internet of things journal, vol. 5, No. 4, august 2018, pp. 2483-2495. DOI: 10.1109/JIOT.2017.2767291

[18] W. Ejaz, M. Naeem, A. Shahid, A. Anpalagan, and M. Jo, "Efficient Energy Management for the Internet of Things in Smart Cities". IEEE Communications Magazine, January 2017, pp. 84-91. DOI: 10.1109/MCOM.2017.1600218CM

[19] J. L. Espinosa-Aranda, N. Vallez, C. Sanchez-Bueno, D. Aguado-Araujo, G. Bueno, O. Deniz, "Pulga, a tiny open-source MQTT broker for flexible and secure IoT deployments". 1st IEEE Workshop on Security and Privacy in the Cloud, Florence (Italy), September 30, 2015

[20] IBM, “Qualities of service provided by an MQTT client”. [Online]. Available: https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_8.0.0/com.ibm.mq.dev.doc/q029090_.htm [Accessed February 8, 2019]

[21] 20-Nov-2018 OpenSSL 1.1.0j is now available, including bug and security fixes [Online]. Available: https://www.openssl.org/ [Accessed January 4, 2019]

[22] Damaševičius, R., Napoli, C., Sidekerskienė, T., & Woźniak, M. (2017). IMF mode demixing in EMD for jitter analysis. Journal of Computational Science, 22, 240-252.