Extended Model of Secure Communication for Embedded Systems with IoT and MQTT

Peter Peniak
University of Žilina,
Faculty of Electrical Engineering, Department of Control and Information Systems
Žilina, Slovak Republic
peter.peniak@fel.uniza.sk

Mária Franeková
University of Žilina,
Faculty of Electrical Engineering, Department of Control and Information Systems
Žilina, Slovak Republic
maria.franekova@fel.uniza.sk

***Abstract* – This article deals with a model of secure communication for embedded devices via Internet of Things. The main focus is paid on MQTT application protocol, which is well known and broadly used. New extended model is based on the new cryptographic methods and optional additional measures on the application layer. These measures, in addition to standard MQTT protocol features, can contribute to reach the required level of secure communication.**

Keywords -IoT;SSL/TSL;MTTQ;AMQP;WSN;Cloud

I. INTRODUCTION

Embedded systems are devices that have intelligence 'embedded' into them using micro-controllers with associated peripherals and can communicate each other via a communication network. Embedded systems, however, have very constrained memory and often use specialized processors. Internet of Things is the network of interconnected devices embedded with electronics, software, sensors and network connectivity, which enables to collect and exchange data between devices via Internet [1], [2].

IoT uses several applications protocols, which are designed for independent communication of smart embedded devices. It offers device to device (D2D) or device to server (D2S) communication model. D2D is based on "Publish/Subscribe" model, from publishers (producers) to subscribers (consumers). D2S has a central broker instance or classical Request/Response protocol. An example of complex IoT implementation is shown by Fig. 1, with selected application protocols according to Table I. Constrained wireless sensors are connected via MQTT-SN protocol and gateway with central MQTT broker instance that is used as a "global data space" and accessible to all applications, or via AMQP gateway to other servers or Cloud [4].

TABLE I. IOT -SELECTED APPLICATION PROTOCOLS

No	IoT protocols		
	Protocol	***Class***	***Remark***
MQTT	Message Queue Telemetry Transport	D2S/ TCP	Broker based
MQTT-SN	MQTT for Sensors Networks	D2S/ WSN	MQTT gateway
AMPQ	Advanced Message Queuing Protocol	S2S/ TCP	Messaging with Queuing

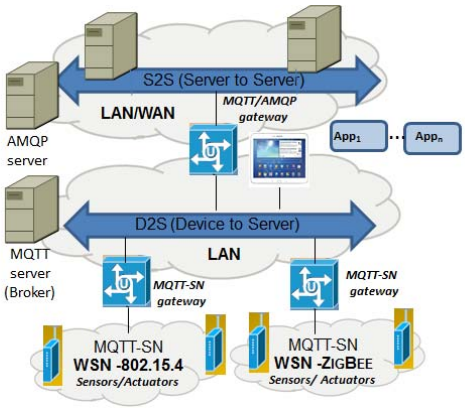

Figure 1. Example of IoT implementation

MQTT is the one of the most used application protocols for IoT, therefore next chapter will provide deeper explanation of protocol features.

II. MQTT APPLICATION PROTOCOL

MQTT (Message Queuing Telemetry Transport) is an application protocol defined by ISO/IEC PRF 20922 standard. It works on top of the TCP/IP protocol stack as a lightweight broker-based publish/subscribe messaging protocol. It was designed to be open, simple, lightweight and easy to implement [13]. This protocol is based on "Publish/Subscribe" model with central "Broker instance", which creates the concept of a "global data space". All communication is represented as reads and writes to the global data space. The broker based publish/subscribe model is composed of:

- Broker is a server, which defines data variables called "Topics" and routes all the messages among connected clients. Topics are managed in hierarchical form (*/root/level$_1$/../level$_n$/Topic$_i$*).
- Publisher is a client that publishes a message to one (or many) topic(s) in the broker.
- Subscriber is a client that subscribes to one (or many) topic(s) in the broker and receives all the messages sent from the publisher.

MQTT protocol architecture is illustrated by Fig.2.

ISBN 978-80-261-0722-4, © University of West Bohemia, 2018

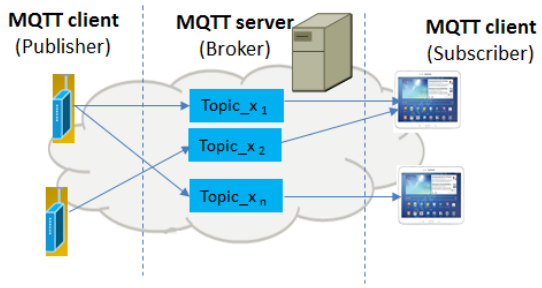

Figure 2. MQTT architecture

The protocol message format is based on fixed header and an optional header with data payload and UTF-8 coding, as shown by Fig. 3. Protocol provides various messages that are used for specific purposes, such as a creation of connection, subscription to topic, or publishing to/from topics. The semantic of message exchange between client and MQTP broker, is illustrated by Fig. 3 as well.

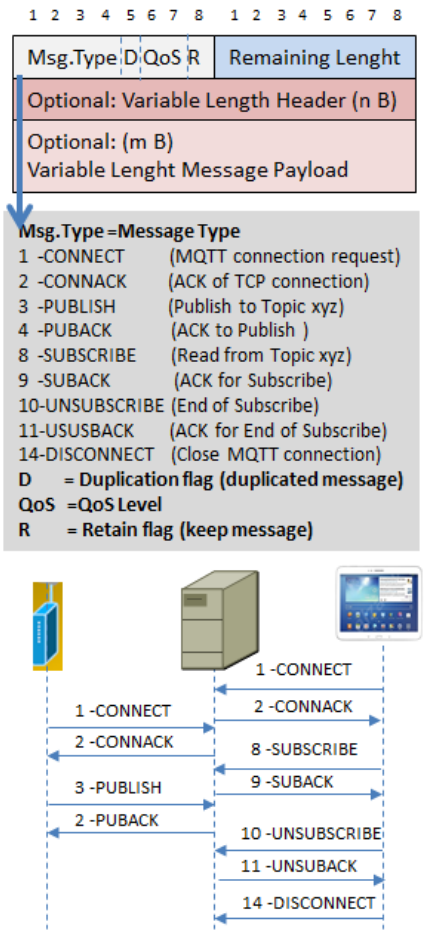

Figure 3. MQTT message format -header

MQTT-SN is the specific version of MQTT protocol for wireless sensor networks (WSN), with the aim of extending the MQTT protocol beyond the reach of TCP/IP infrastructure for Sensor and Actuator solutions [12].

III. NEW TRENDS OF SECURE COMMUNICATION

The communication of embedded devices via Internet, however, brings the new challenge with respect to a cyber-security. In order to withstand malevolent attacks, the end-to-end communication channels must be secured using cryptographically strong encryption and authentication algorithms [11]. Generally, the network communication is secured by standard protocols in Layers 2-7 (RM-OSI), as shown in Table II. IPsec protocol is mostly used for an encryption of IP packets and creation of secured VPN (Virtual Private Networks) connections through Internet. SSL/TLS (Secure Socket Layer/Transport Layer Security) protocols use a record structure to encrypt and authenticate the application data and to embed them into a TCP/IP stream.

TABLE II. PROTOCOLS FOR SECURE COMMUNICATION

No	Security protocols	
	Protocol	***Remark***
Layer 7	S/MIME, Http digest,.	Encryption/Authentication in application layer
Layer 4	SSL, TLS, WTLS,..	Encryption/Authentication in Socket layer
Layer 3	IP Sec	Encryption/Authentication, application independent.
Layer 2	WPA2, A5 (GSM),…	Link layer encryption

The biggest obstacle is the limited memory and the restricted processing power available in embedded systems. The first topic is the generation of true random material in a low-entropy environment and the second item will concentrate on the computationally intensive operations required by the standard RSA public key cryptosystem as used by IPsec's Internet Key Exchange protocol, the SSL/TLS handshake protocol, and the SSH public key-based authentication [5]. Using its RSA private key an SSL/TLS server running on an embedded system must compute a singular modular exponentiation in order to decrypt the pre-master secret that had been previously encrypted by the SSL/TLS client with the server's RSA public key [6]. To address this issue and to provide feasible technology for encryption, the new LWC (Lightweight cryptography) methods are under development. Lightweight cryptography has been one of the "hot topics" in symmetric cryptography in the recent years. A huge number of lightweight algorithms have been published, standardized and/or used in commercial products. The most promising is an effort within preparation of the new block/streams ciphers with using of a secret key [10]. Development of LWC block ciphers is based on modifications of existing block cipher from area of conventional cryptography as DES (Feistel structure) or AES (permutation and substitution structure). In contrast to current standards, the following main principles have been applied:

- smaller blocks (64 b/ 80 b),
- shorter length of keys (80 b, resp. 112 b),
- simplified rounds (prefer 4 bits S-boxes),
- simplified generation of keys in rounds,
- minimal implementation (just encryption).

ISBN 978-80-261-0722-4, © University of West Bohemia, 2018

On the base of these characteristic, the following new algorithms were development (for RFID tags) [8], [9]:

- LWC PRESENT (on the base of AES),
- LBLOCK (on the base of DES),
- TRIVIUM (stream based),
- GRAIN (stream based).

TABLE III. CHARACTERISTIC OF SEVERAL TYPES OF CIPHERS FROM AREA OF LWC

Name	Length of key [b]	Block size [b]	Number of cycles per block	Area [GE]	Speed [k bit/s]
Block ciphers					
PRESENT	80	64	32	1571	200
LBLOCK	80	64	-	1320	200
AES-128	128	128	1032	3400	12
Stream ciphers					
Trivium	80	1	1	2599	100
Grain	80	1	1	1294	100

With using the new algorithms, hardware and resource requirements can be significantly reduced, which would enable an encryption for communication of embedded devices. In addition, the overall security can be extended by application layer:

- Obligatory authorization of embedded systems (user-id/password, device-id).
- Activation of encryption, if supported by used application protocols.
- Access restriction for global data space (topics, items), based on user-id/device-id.
- Extended security measures on server side, filtering based on IP, ports.
- Extension of security directly on application level with redundancy and validation.

IV. PROPOSAL OF EXTENDED SECURITY MODEL AND OBTAINED RESULT

MQTT protocol supports a generic server authorization, based on provided user name with password from client system. This is an optional protocol feature, although should be set as a mandatory for all embedded devices. The broker must reject any "CONNECT" command with wrong username/password or client ID, by sending "CONNACK" message, as shown on Fig. 4 [14].

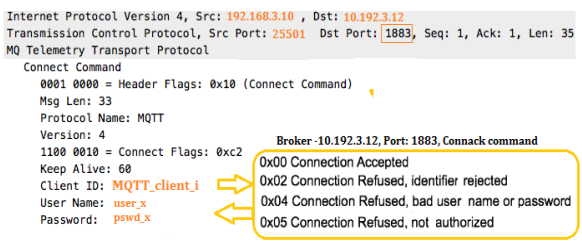

Figure 4. Example of MQTT message

If an embedded device cannot use encryption due to limited resources, the authorization data are directly visible as a text in "Connect" command message, what is a potential security issue.

The most of constrained devices are not able to afford even that simple security level. Therefore, an extension of standard application model is proposed to increase security level. The essential idea is based on possibility to use modified application logic with redundancy. Extended model is based on additional application node, called "Validator", which is proposed to provide validation of published values, based on functional transformation of provided values and comparison of the result with an expected pattern, as shown on the Fig. 5.

The ordinary MQTT device uses an assigned $Topic\_x_i$ in order to publish value $x_{i,}$, as shown by Fig. 5a). On the other hand, MQTT device with extended security model would have to have two additional topics ($Topic\_x_{i2}$, $Topic\_x_{i3}$), which are assigned to primary topic ($Topic\_x_{i1}$) with an additional validator Vx_i, as shown by Fig. 5b) and expressed by (1).

$$Topic\_x_i \sim \{Topic\_x_{i1}, Topic\_x_{i2}, Topic\_x_{i3}, Vx_i\} \quad (1)$$

Where:

$Topic\_x_{i1}$	- masked original value,
$Topic\_x_{i2}$	-parameter for masking,
$Topic\_x_{i3}$	-validated value for publishing,
Vx_i	-assigned validator entity.

An application logic of constrained MQTT device applies a simple transformation function f_i, which performs masking of original value (x_i) with predefined parameter (x_{i2}), resulting in transformed value (x_{i1}), as expressed by (2).

$$x_{i1} = f_i(x_i, x_{i2}) \quad (2)$$

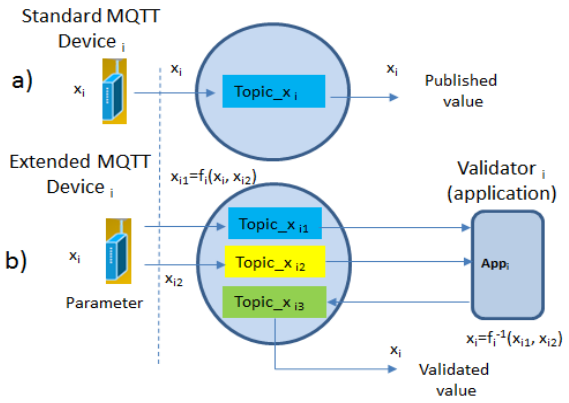

Figure 5. Extended security model with "Validator" entity

The both values are published to variables *Topic_x*$_{i1}$, *Topic_x*$_{i2}$. Assigned validator *Vx*$_i$ knows used transformation function f_i associated with *Topic_x*$_{i1}$. The original value (x_i) can be obtained based on inversive function $f^{-1}{}_i$ and written to variable *Topic_x*$_{i3}$, as expressed by (3). The both published values (x_{i1}),(x_{i2}) are in addition checked if they match by validation function F_i. If not, *Topic_x*$_{i3}$ would be set to "0", as expressed by (4) and not validated.

$$x_{i3} = f^{-1}{}_i(x_{i1}, x_{i2}) = x_i \quad (3)$$

$$x_{i3} = F_i\{x_{i1}, x_{i2}\} = 0 \quad (4)$$

The proposed extended model has been simulated in test environment with WEB based emulation of MQTT Broker instance, validator entity and emulation of the both generic and extended MQTT device, as shown by Fig. 6. The extended security model has been proven and validator was able, based on provided parameters and transformation function, to calculate validated value and confirm its correctness, as shown by Fig. 6a). On the contrary, the standard MQTT device did not get validation confirmation after trying to publish value, as expected and shown by Fig. 6b).

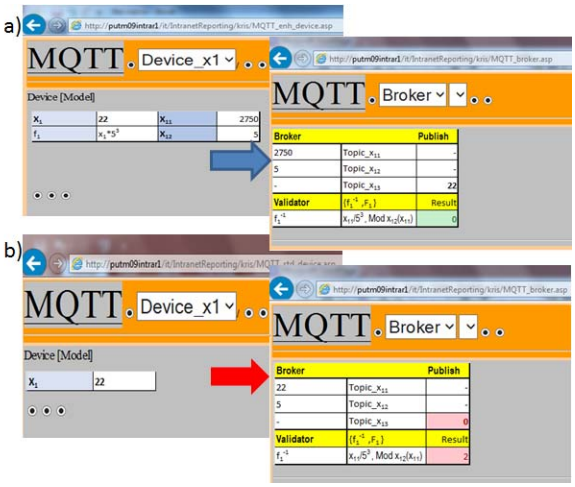

Figure 6. Simulation of extended security model with "Validator"

V. Conclusion

As explained, IoT security features with MQTP protocol are not strong enough, mainly due to just optional authorization and lack of encryption power in embedded systems. Till the new cipher algorithms are made available, there is still a risk that two most dangerous security issues might happen, such as a creation of false command to actuator, or provisioning of false data instead of real data from sensor. In addition to standard MQTT authorization and access restrictions, the risk can be at least partially, compensated by redundant application logic with using of proposed extended security model.

The proposed additional application node (Validator) is capable to validate if published data are correct, based on paired Topics with a parameter prefix, modified value and assigned transformation function. The overall result of validation is either the original value, or confirmation of a command. The proposed model was tested on WEB based simulation platform and its enhanced security features has been proven. After simulation, the proposed solution will be applied for constrained devices and MQTT Broker will be extended by Validator as the second step. The same method can be applied for execution of commands.

Acknowledgment

This work has been supported by the Educational Grant Agency of the Slovak Republic (KEGA) Number: 016ŽU-4/2018 Modernization of teaching methods of management of industrial processes based on the concept of Industry 4.0.

References

[1] E. Bubeníková, P. Bubeník, "Internet vecí (IoT) = Internet of things (IoT) ", Technológ. Vol. 9, No: 1 (2017), pp. 131-134, ISSN 1337-8996

[2] P. Papcún, , I. Zolotová, "IoT household controlled by cloud technology". In: International Journal of Internet of Things and Web Services. Vol. 1 (2016), p. 103-109. - ISSN 2367-9115, 2016

[3] J. Mroz, "Shared Data of the Internet of Things and in Idustry 4.0," Detecon Management Report BLUE, Vol. 1 2015,pp.24-27. In: http://www.detecon.com/ap/ap/files/DMR_blue_IoT_Industry_4_0_Mroz_01_2015_E.pdf

[4] P.Peniak, "Cloud computing and integration manufacturing information systems with process control". In: Slovak. Žilina, 2014. Inaugural dissertation work. University of Zilina.

[5] W. Stallings, "Cryptography and networks security". In: https//williamstallings.com

[6] M. Franeková, K. Rástočný, "Kryptografia v bezpečnostne relevantných systémoch", Edis ŽU v Žiline, 2017, ISBN 978-80-554-1310-5

[7] CRYPTREC: Cryptography Technology Guideline. Lightweight Cryptography Group, 2017

[8] NIST IR 8144: Report on Lightweiht Cryptography. In: https://doi.org/10.6028/NIST.IR.8144

[9] William J. Buchanan at all.: Lightweight Cryptography Methods. In: Jornal of Cyber Security Technology. May 2018. In: https://doi.org/10.1080/23742917.2017.13849147

[10] A. Bogdanov et al., ''PRESENT: An Ultra-Lightweight Block Cipher,'' Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES 07), LNCS 4727, Springer, 2007, pp. 450-466.

[11] A. Steffen, ''Secure Communications in Embedded Systems'' Zürcher Hochschule Winterthur, CRC Industrial Information Technology Handbook, 2006. In: https://pdfs.semanticscholar.org/a557/61915023c8904cfedd6dd95b9aff825dbc59.pdf

[12] P.R.Egli, „MQ Telemetry Transport -An Introduction to MQTT, A protocol for M2M and IoT Applications". 2016, In:http://www.indigoo.com/dox/wsmw/1_Middleware/MQTT.pdf

[13] R.K.Barde, U.R.Gandhi, "Study of Message Queue Telemetric Transport", International Journal of Advanced Innovative Technology in Engineering (IJAITE), Vol. 1, Special Issue 2, July-2016, ISSN: 2455-6491, pp.1-2

[14] V.Pasknel, „Hacking the IoT with MQTT", Morphus Labs, 2017, In: https://morphuslabs.com/hacking-the-iot-with-mqtt-8edaf0d07b9b