SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 17, No. 3, October 2020, 389-403

UDC: 004.056.5:004.738.5 **DOI: https://doi.org/10.2298/SJEE2003389D**

Authentication Algorithm for Internet of Things Networks Based on MQTT Protocol

Dmitrii Dikii[1]

Abstract: The authentication algorithm for machine-to-machine communication devices via the MQTT Protocol is considered, which allows verifying the device legality without sending the password. The algorithm protects the Internet of things network from unauthorized access, ensures confidentiality by generating a common session key, and provides protection against attacks of the "man in the middle" and others. The obtained experiment results showed noticeable performance improvement compared to the TLS Protocol. The cryptographic strength of the proposed algorithm is based on the discrete logarithm problem. The main advantage of the algorithm is the ability to authenticate in one request-response cycle.

Keywords: Authentication, MQTT, Zero-knowledge, Key management, Internet of things.

1 Introduction

Industry 4.0 development and the implementation of cyber physical systems into people's daily lives have raised the question of implemented technology security. One of the most popular solutions in automation is the implementation of the Internet of Things (IoT). It includes such concepts as Smart House and Smart City [1]. The main difference of the equipment used in the IoT from the common one is that it is capable of making decisions based on the input data and the communications linkage via the Internet, although it has limitations in computational performance and energy storage of the independent power supply source. These aspects impose multiple limitations. Various light-weighted communication protocols are being developed to solve the problem of economic data transmission between multiple devices. For example, the CoAP (Constrained Application Protocol) algorithm [2], the architecture of which resembles that of the HTTP (HyperText Transfer Protocol), and MQTT (Message Queue Telemetry Transport) protocol that employs the publish-subscribe model [3] were presented in 2013. XMPP (Extensible Messaging and Presence Protocol) [4] and DDS (Data Distribution Service) [5] protocols were

[1]Faculty of Secure Information Technologies, Saint-Petersburg National Research University of Information Technologies, Mechanics and Optics, 197101, St. Petersburg, Russia; E-mail: dimandikiy@mail.ru

presented as well. Those protocols operate on the protocol suite TCP/IP. Another course of the IoT development is creation of lower level protocols of the OSI model. The most well-known is the suite of the protocols ZigBee and LoRaWAN [1, 6]. The information security in the mentioned protocols differs in its organization. The mechanism, present in each protocol, is the integrity control in the form of the CRC (Cyclic Redundancy Code) protocol. The symmetrical e-d algorithm AES-128 bit (Advanced Encryption Standard) [7-9] performs confidentiality control. Protocols of higher levels provide asymmetrically and symmetrically transmitted data encryption, user authentication for information exchange, and integration control [10]. Generally, security functions require additional computational and energy costs. They also demand a setup of higher requirements for non-volatile energy-dependent memory of the execution units. This paper suggests an algorithm that enables to authenticate devices for M2M (Machine to Machine) connectivity without transmitting key information via exposed paths. Moreover, it enables to generate a session key for further symmetrical encryption of data blocks. The given protocol is completely compatible with MQTT, one of the most widespread protocols of the IoT, which makes it possible to compare the suggested algorithm with the existing information security methods.

2 Related Work

According to the review presented in [11] the botnet network Mirai has challenged the world society in the context of cyberattacks, when devices gained unauthorized access to other devices to execute a malicious code. The authors pointed out the main security mechanisms that must be employed to prevent such incidents. One of the most significant mechanisms is authentication. Thus, the development of authentication protocol adapted to the environment of the IoT is of current interest. The TLS (Transport Layer Security) protocol is often used over the MQTT protocol. Fremantle et al suggest employing the 0AUth2 protocol [12]. Another focus area in device authentication is the development of algorithms based on the zero-knowledge algorithm, presented by Fiat and Shamir [13]. In [14] the authors suggest an algorithm for M2M authentication where a device graph with a table of MAC (Media access control) addresses is created as a key field. The generation of the session key is based on the Diffie–Hellman algorithm. This algorithm is suitable for an all-to-all network architecture. A multi-graph zero-knowledge-based authentication system in the IoT is suggested in [15]. This method is based on graph construction, which requires an enormous amount of message exchange rounds between devices [16]. Singh et al [17] present a different approach to authentication based on asymmetric encryption on elliptic curves. However, the calculations based on this approach are labour and resource intensive. Niruntasukrat et al suggest separate servers for authentication and authorization

[18], which is not always possible. Han et al present a light-weighted authentication protocol of the IoT based on the symmetric block-encryption algorithm [19]. Such an algorithm is more suitable for "point-to-point" authentication. Bhawiyuga et al [20] suggest a method based on security tokens with the invocation of authorization and authentication.

In the article [21] authors propose to use password authentication scheme for M2M communication on the application level. It is based on one-way hash functions, XOR (Exclusive or) operations and a symmetric encryption algorithm. The authentication procedure consists of two request-response cycles. The private keys for encryption should be preinstalled.

A blind signature is observed as a method of authentication for the IoT networks in [22]. The authentication scheme includes an additional storage server that contains signatures. To authenticate through that method four request-response cycles should be done: two between the gateway and the device and two between the device and the trusted storage server.

The authentication algorithm presented by Abdulrahman et al uses only symmetric encryption and hash functions without third parties. However, the secret keys for encryption should be pre-installed on the device side and the gateway side. This key must be unique and stored in a nonvolatile memory storage. The authentication procedure consists of four messages or two request-response cycles [23].

A lightweight algorithm for M2M authentication was presented by Esfahani et al. It uses only hash functions and XOR operations. To authenticate the device and the gateway between each other they transmit only three messages. But this scheme involves an authority center [24].

Due to the fact that M2M communication is widespread, the question of secure and lightweight authentication is becoming relevant presently. The authentication procedure is easier to implement at the application layer of the OSI model. The previously discussed methods allow authentication without the use of asymmetric encryption algorithms and the involvement of third parties, certification authorities, etc. However, they all need two and more stages of request-response cycles, which cannot always be implemented within the existing application protocols, such as MQTT. This protocol provides a one-cycle request-response authentication password scheme using CONNECT and CONNACK messages.

The purpose of this article is to present the developed secure authentication scheme for M2M communication, that may be integrated into MQTT networks and be compatible with it (have one request-response authentication cycle).

The given algorithm must be able to generate a common session key for further usage in symmetrical block and stream encryption. Its main difference

from the mentioned algorithms is that it is able to authenticate in a single request-response cycle. This allows it to use authentication through the MQTT protocol and makes it less energy costly. Moreover, the algorithm must be compatible with the existing systems without disturbing their functionality.

3 Algorithm

The MQTT protocol has a “publish-subscribe” architecture model that consists of two main components. The first one is the end-point devices, which perform the communication traffic. The second component is the gateway. Its function is the logistics of input and output messages. Thus, the messages, delivered to the gateway, are redirected to the intended device or devices. The given algorithm is convenient for multicasting. For example, the publisher device generates one output message for n subscribers. Then, each generated message is marked with a Topic. In order to receive a message with any given topic the subscriber or receiver device subscribes to the topic.

Two mechanisms provide security in the given protocol. One is authentication by username and password, the other is access control by the ACL (Access control list) file or database. The password and username are transferred explicitly in the body of the CONNECT message (the MQTT protocol supports 16 types of messages [3]). If the authentication is passed, the gateway replies with the message CONNACK correspondingly.

As presented in [1, 11, 12, 25, 26], it is suggested to employ the TLS protocol to provide secure authentication between devices and the gateway. This protocol has two stages. The first is the generation of a common session key based on certificates of the X509 format via Diffie-Hellman protocol (DH - Diffie-Hellman [27] or ECDH – Elliptic curve Diffie-Hellman [28]). The second is the employment of the symmetric or asymmetric encryption algorithm, e.g. AES, RSA (Rivest–Shamir–Adleman), ChaCha20 and a hash-function, e.g. SHA256 (Secure Hash Algorithm) [29]. This approach has several disadvantages. Firstly, the chain includes a third party – certification authority, which can verify the authenticity of the issued certificates for the devices and the gateway. Secondly, the devices must store their certificates, private keys and the gateway’s certificate on a non-volatile storage. Thirdly, the cypher suites employed in the gateway must be compatible with the devices. If they are not compatible, it is impossible to create a secure data transmission link. The most important advantage of the given method is full link encryption. In order to connect via TLS protocol, the device must contact the gateway by the latter address, but with a different port number. For example, for an unsecured connection the address would be «tcp://127.0.0.1:1883», whereas for a connection via TLS protocol it would be «ssl://127.0.0.1:8883». Thus, a secure connection is made via TLS protocol. After that, authentication on the gateway

is done against the username and password. To be noted, the employment of an asymmetric encryption is more calculation costly. Consequently, the developers try to minimize the asymmetrical encryption in the IoT.

The suggested algorithm is based on zero-knowledge protocols – the Schnorr algorithm [16] and the algorithm presented in [30]. Zero-knowledge algorithm was originally presented in the end of the 20th century. The main goal of this algorithm for the server (gateway) is to make certain that the client (device) knows the password without directly transmitting it. The difficulty of employing those protocols through MQTT protocol is caused by the fact that only a single request-response cycle is used to authenticate on the gateway side.

3.1 Registration

Abbreviations from **Table 1** will be used further in the text.

Table 1
Notations.

Symbol	Description
password	A secret user's password as a combination of symbols
login	Unique identity of user
PSK	Private secret key of the gateway
g, *p*	Public large numbers
cl_id	Device identity
H(.)	One-way hash function
||	Concatenation operator
⊗	XOR operator
mod	Modulo operator
$E_K()$	Symmetric encryption on key K
F	A secret unique token for a device
R	Random large number
timestamp_i	Timestamp of a message
salt	A random set of symbols of a certain length

The end-point device has to perform the registration procedure through a secure channel. The device should provide values login and *cl_id* to the gateway. Then the gateway calculates token *F*:

$$F = H(H(\log in \,||\, cl\_id) \otimes PSK), \tag{1}$$

where the variable *PSK* is a gateway`s secret private key. The gateway sends to the device *F*, *g*, *p*. Parameters *p* and *g* are public and accessible to all. Generally, *p* is a large prime number, *g* is less than *p*. Parameter *p* should be at least 512 bits. It is necessary for number *q* to exists, which is the multiplier of a number *p*–1. Parameter *g* has to meet the condition $g^q \equiv 1 \bmod p$. Those conditions make the algorithm cryptographically strong [31].

After the device received the response with p, g, F values from the gateway it can calculate values X and Y, correspondingly:

$$X = H(password), \tag{2}$$

$$Y = g^X \bmod p. \tag{3}$$

Values X, Y, F, g and p should be stored in nonvolatile device memory. The last step of registration is transmitting value Y to the gateway. The gateway will store value Y. The registration procedure is finished. It can be released through, for example, a web-application. The registration procedure is shown on Fig. 1.

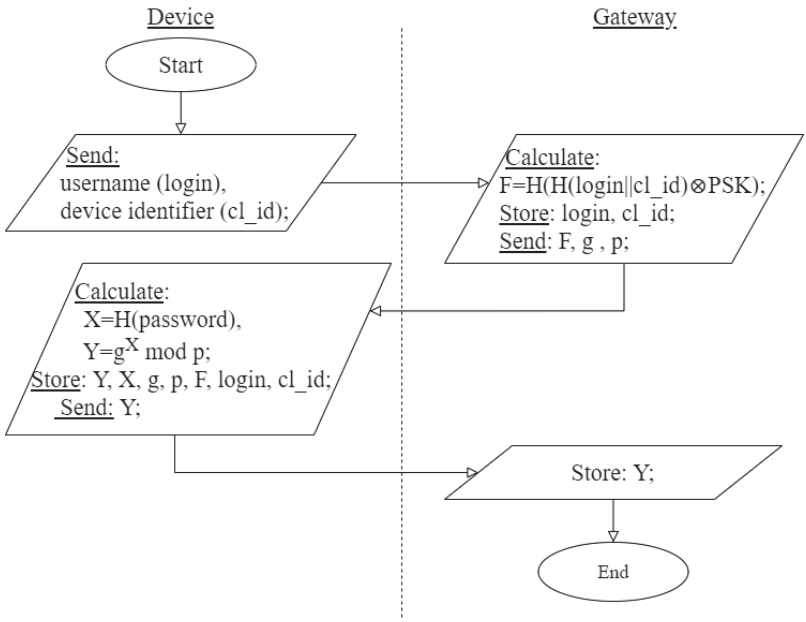

Fig. 1 – *Registration procedure of the proposed algorithm.*

3.2 Authentication

The authentication scheme is presented on Fig. 2. The end-point device has X, Y, F values and public g, p values. To pass the authentication procedure the device generates a large random number R and calculates:

$$T = g^R \bmod p. \tag{4}$$

Using T and F values the end-point device generates a session key as:

$$K = F \otimes H(T). \tag{5}$$

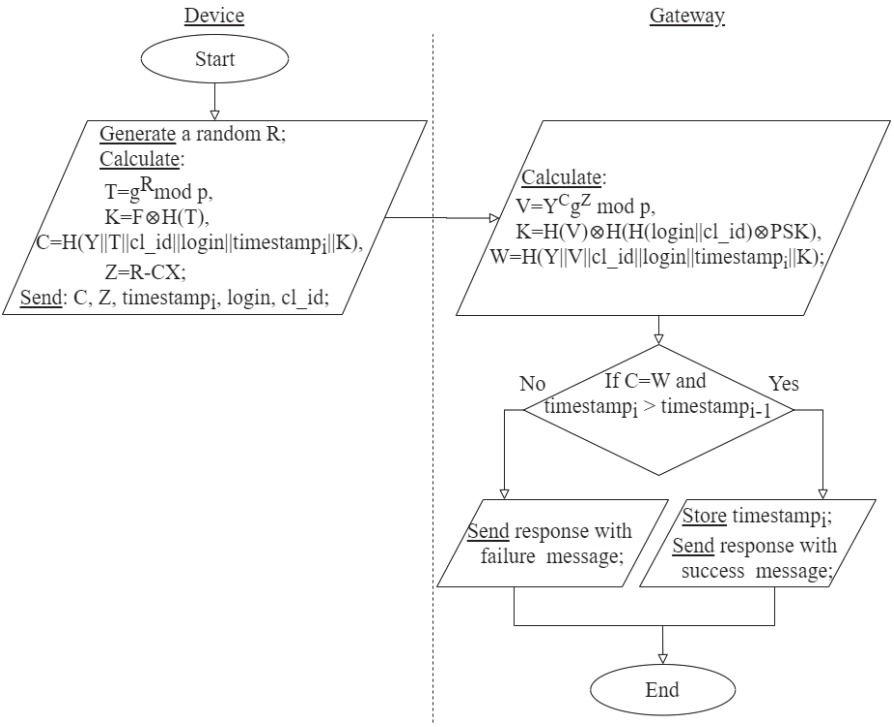

Fig. 2 – *Authentication procedure of the suggested algorithm.*

In the next step, the connection time must be registered in milliseconds as a variable $timestamp_i$. This variable is required to avoid reauthentication by the same CONNECT message. The gateway will later check the timestamp, device identifier and username. If the timestamp is outdated (the message with the same content was received earlier), the end-point device will be denied in authentication. The next step is to calculate variable *C,* which is the hash value of parameters *Y*, *T*, client identifier *cl_id*, $timestamp_i$ and session key *K*:

$$C = H(Y \,||\, T \,||\, cl\_id \,||\, \log in \,||\, timestamp_i \,||\, K)\,. \tag{6}$$

Number *Z* is then calculated:

$$Z = R - CX\,. \tag{7}$$

The end-point device sends a CONNECT message with the login value into the username field and values *C*, *Z*, $timestamp_i$ into the password field. The data about the client identifier is always sent in the variable header of the CONNECT message of the MQTT protocol, thus there is no necessity in information repeating, but the presence of the hash value *C* conclusively verifies the message with the end-point device. This is necessary to secure from message spoofing.

The gateway that received the CONNECT message, finds the user by username in the database. Then it uses the stored variable Y to calculate variable V by:

$$V = Y^C g^Z \bmod p \tag{8}$$

After calculating value V, the gateway can generate a session key on its side using the equation:

$$K = H(V) \otimes H(H(\log in \,||\, cl\_id) \otimes PSK) = H(V) \otimes F\,. \tag{9}$$

The next step is to calculate value W, according to the equation:

$$W = H(Y \,||\, V \,||\, cl\_id \,||\, \log in \,||\, timestamp_i \,||\, K) = \mathrm{C}\,. \tag{10}$$

To prove the legality of the end-point device on the gateway value C of the CONNECT message is compared with the calculated value W. It also verifies whether the timestamp is outdated or not. If the timestamp is up-to-date, it is registered. To prove the validity of cryptographic transformations values Y and Z are put into (8), which results in:

$$V = g^{XC} g^{(R-CX)} = T\,. \tag{11}$$

Parentheses removed, one gets $V=g^R \bmod p$, which represents variable T from the device-side. Thus, the gateway using equation (10) can calculate hash value C, consisting of Y, T, $timestamp_i$, cl_id and K.

To insure that the gateway communicates with an authenticated device and checks the equality of variables C and W, which consist of the login, cl_id, $timestamp_i$, transformed PSK and password values.

On the other side, the end-point device must identify the gateway. It can be accomplished by sending a message from the gateway to device. But a CONNACK message of the MQTT protocol does not have any field to input additional data. The CONNACK message has limited size fields in the fixed and variable headers, each of them is no more than a few bits. The payload field is missing. Thus, this type of message is incompatible as a data transmitter.

The method of gateway approval on the device side is presented below. The additional header is attached to the first message from the gateway to the device that contains a payload field. At the beginning of it, a username is included with the addition of a random symbol pattern of a certain length (salt) encrypted on the common session key. The length of the random pattern must be pre-arranged. For example, in the algorithm employment, the following method was used: as long as the division excess of the username length by eight is not equal to zero, a random symbol is to be added at the username's end. If the length of the username is a factor of eight symbols, then eight random symbols are to be added at its end. This will enable to get varieties of encrypted texts in case of a block encryption on a single username and common session key. Setting the

username at the beginning will allow the device to become certain that the gateway actually has the common session key. If the username is not found at the beginning of the message during decryption, the message will not be processed further. To make integrity control of the message we can add a hash value of the message in the same manner.

The example of the first message payload format from the gateway to the end-point device:

$$M = E_K(\log in \,||\, salt \,||\, message) \tag{12}$$

Cryptographic robustness of the transmitted messages depends on the robustness of the block and flow cypher algorithms.

The advantages of the developed algorithm are that the authentication takes place during one request-response cycle, there is no need for a third party, nor to store certificates on a nonvolatile storage, comparing to the TLS protocol. The disadvantage of the algorithm is the device configuration. Such system parameters as p, g and F must be known before the connection. The block/flow encryption algorithm has to be pre-arranged, because the usage of various cypher algorithms will lead to the malfunction of the proposed algorithm.

Such mechanisms, like setting the username with a random symbol pattern at the beginning of the message, and setting a timestamp, zero-knowledge authentication and message hash value adding, eliminate a lot of threats.

4 Attack Scenarios

1. **Brute force.** An adversary may try to guess the password. To avoid such a situation the legal user of the IoT network has to choose a secure password. It can be reachable by a password policy on the gateway side during the registration step. However, the adversary does not know the value of F, which depends on the secret key PSK of the gateway. Without value F, the session key cannot be properly generated and the computation of value C will not be valid which will be checked on the gateway side. Thus, the adversary has to guess both: the password and value F.
2. **Replay attack.** The adversary may intercept a CONNECT massage transmitted from the device to the gateway and resend it later to authenticate himself. At first, the timestamps will be checked on the gateway side. Even if the adversary will change the timestamp value, value C should be changed too. But the adversary does not have enough information, such as X, F to generate value C properly. In addition, every new session will have a new session key because value R is randomized for every new connection.

3. **Man-in-the-middle attack.** To execute such an attack the adversary has to obtain the session key. As the session key is not transmitted on an unsecure channel, it must be generated. From this side the adversary does not know values Y or X and F, and it is not transmitted over the channel. Thus, the suggested algorithm is resistant to that attack.
4. **Device proof on the gateway side.** The gateway can be certain of the device legality by checking the equality of values W and C by the zero-knowledge algorithm. It shows the identity of the pair password-login. On the other hand, value C contains the session key generated on value F. According to equation (9) if value W is equal to value C then the device used the same F for the session key generation that was generated from login, *cl_id* and *PSK* on the gateway side.
5. **Gateway proof on the device side.** A CONNACK message of MQTT protocol is not possible to be modified due to protocol specification. If it is the adversary's gateway, it cannot get enough information from the CONNECT message to obtain the session key or authentication. All the following information from the device will be encrypted and confidentiality will be kept. The first message with payload field from the gateway to the device must be modified by attaching the additional header (equation (12)). The device can check the gateway legality by decrypting the message and extracting an additional header. If there are any errors while decrypting or the additional header differs from expected ones that indicates failure of generating a session key on the gateway side.
6. **Gateway database leak.** If the adversary got access to the database on the gateway side, he can retrieve login and Y pairs. Values X, F should not be stored in the database. The adversary cannot get value X from Y due to the discrete logarithm problem. However, without X value it is impossible to be authenticated on this gateway according to equation (7). Furthermore, the adversary cannot obtain the session key because the *PSK* value is still unknown to him.
7. **Modification attack.** Almost all values that are transmitted through a CONNECT message are protected by one-way hash function inside value C. Thus, if modifications were made on the message the equality of values C and W, as results of a hash function on the gateway side, will not be reached.
8. **Impersonation attack.** If another device will try to connect to the gateway, it must have login, *cl_id*, password and F values. The last depends on the *PSK* value of the gateway. Thus, the suggested algorithm is resistant to that attack.

5 The Suggested Algorithm and TLS Protocol Comparison

An experiment has been conducted in order to compare the proposed algorithm and the TLS protocol using the following method. The MQTT gateway software with an open source code has been modified [32]. Microcomputer Raspberry Pi 3 model B was used as a gateway with the following specifications: CPU (Central processing unit) - ARM Cortex-A53, processor speed 1,2 GHz, RAM (Random-access memory) 1 GB. The costliest process is the connection of the device to a gateway. Consequently, calculations were carried out concerning the time spent on the connection via:

– an insecure channel;

– a channel secured by the TLS protocol;

– a channel secured by the suggested algorithm (numbers p, g, R are not less than 512 bits);

– a channel secured by both the TLS protocol and the suggested algorithm simultaneously.

The experiment consisted of five thousand connections on each of the four above mentioned connection types. The results are depicted in Fig. 3.

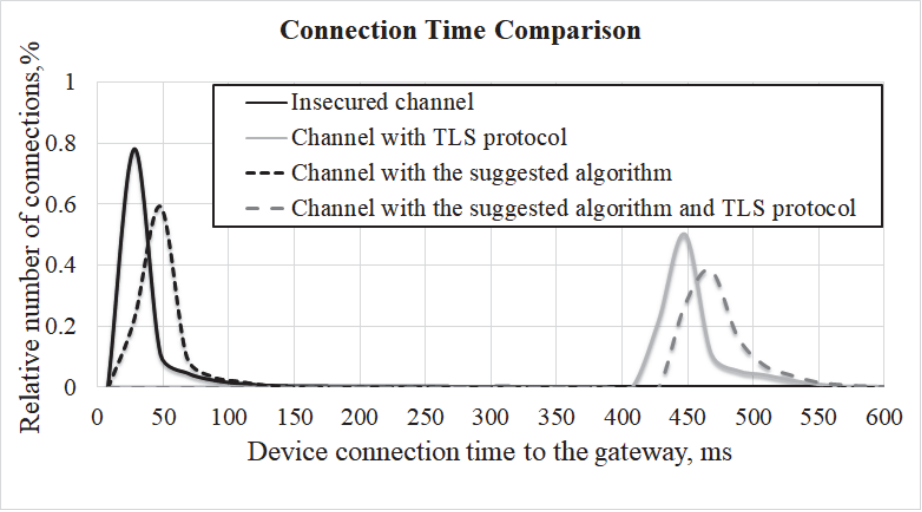

Fig. 3 – *The comparison of time spent on insecure/secure channel connection; the time step of x – axis* – 20 ms.

The disadvantage of the proposed algorithm is the device synchronization over time. The employment of a timestamp presumes that time on the device must be synchronized and cannot turn back. Moreover, the device must store the system parameters (two large prime values p and g) and F value in a constant

non-volatile storage. In addition, the proposed algorithm is cryptographically strong only when prime p is large enough, not less than 512 bits. When transmitting messages, there data redundancy is small. The implemented mechanism into MQTT protocol may be adjusted to check data integrity like TLS protocol. Therefore, the proposed algorithm may raise the level of authentication security for IoT networks.

In the course of the experiment, the author got the statistical parameters of the end-point device connection time to the gateway. The data is presented in **Table 2**.

Table 2

The comparison of device connection time to the gateway using insecure/secure channels.

Authentication type	Insecure channel	TLS protocol	Suggested algorithm	Suggested algorithm and TLS protocol
Minimal time of connection, ms	8	415	19	421
Maximal time of connection, ms	3015	3438	1978	4659
Mean time of connection, ms	36.2	460.0	45.7	480.7
Mean square deviation	104.1	121.8	71.1	156.4

According to the experiment results, the authentication via an insecure channel is the fastest and it has the average mean square deviation. The implementation of the TLS protocol significantly slows the process of authentication. Thus, the average time of the client's connection to the gateway is 12 times slower and takes about half a second. The proposed algorithm has higher execution speed than the TLS protocol: the average time of connection relating to the insecure channel authentication is about one and a half times slower (60% of the connections take less than 50 ms), and ten times faster than through the TLS protocol. However, the proposed algorithm has lower dispersion and mean square deviation, which points to data scattering. The employment of both security mechanisms simultaneously during authentication gave the worst results.

The time of message delivery via secure and insecure channels was estimated in order to achieve the effect of adding a username with random patterns of symbols at the beginning of each message. The experiment showed that there is no obvious influence on the execution speed of message delivery from implementing additional data. The difference equals 1-2 ms.

Thus, the efficiency of the proposed authentication algorithm related to the IoT was illustrated, where the execution speed, energy saving and computation costs are key parameters.

6 Conclusion

This paper presents an authentication algorithm of M2M communication based on the MQTT communication protocol, which enables to verify the authenticity of a device without directly transmitting the password. The most widespread method of data security on MQTT is the creation of a secure connection via the TLS protocol. The disadvantages of such method are the engagement of a third party (certification authority), the certificate storage in a non-volatile memory storage, the inevitable compatibility of cryptographic algorithm sets on the gateway and the end-point device, which provides for additional requirements for the hardware design. The suggested algorithm has several advantages comparing to the TLS protocol, for example:

– No necessity to transmit the password from the device to the gateway;
– Generation of the common session key;
– No necessity to use and store security certificates in the X509 format;
– No dependence on third parties;
– Faster device communication, which is proved by the experiment's results;
– Cryptographic robustness is provided due to the discrete logarithm insolvability within the reasonable amount of time, one-way hash functions and XOR operations;
– Security from the attacks by employing timestamps, additional data in the message payload to prove legality gateway and device for each other, discrete logarithm problem, one-way hash functions and XOR operations;
– Full compatibility with the communication protocol MQTT;
– Generation of the common session key and authentication in a single request-response cycle;
– Guarantee of confidentiality by using the block cryptographic algorithm on different keys for each session.

The disadvantage of the proposed algorithm is the device synchronization over time. The employment of timestamps presumes that time on the device must be synchronized and cannot turn back. Moreover, the device must store the system parameters (two large prime values p and g) and the F, X, Y values in a constant non-volatile storage. In addition, the proposed algorithm is cryptographically strong only when prime p is large enough. When transmitting messages, there is a small data redundancy. The implemented mechanism into MQTT protocol may be adjusted to check data integrity like the TLS protocol. Therefore, the proposed algorithm may raise the level of authentication security for IoT networks.

7 Acknowledgment

The reported study was funded by Russian Ministry of Science (information security) №11/2020.

8 References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash: Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, IEEE Communications Surveys & Tutorials, Vol. 17, No. 4, Fourthquarter 2015, pp. 2347 – 2376.

[2] Z. Shelby, K. Hartke, C. Bormann: The Constrained Application Protocol (CoAP), Internet Engineering Task Force (IETF), Universitaet Bremen TZI, Germany, June 2014.

[3] R. J. Cohn, R. J. Coppen: OASIS Standard Incorporating Approved Errata 01, MQTT Version 3.1.1, OASIS, 2014.

[4] P. Saint-Andre: Extensible Messaging and Presence Protocol (XMPP): Core, Internet Engineering Task Force (IETF), Cisco, March 2011.

[5] Data Distribution Service (DDS) v1.4 Specification, Object Management Group, April 2015.

[6] U. Raza, P. Kulkarni, M. Sooriyabandara: Low Power Wide Area Networks: An Overview, IEEE Communications Surveys & Tutorials, Vol. 19, No. 2, Secondquarter 2017, pp. 855 – 873.

[7] C. Gomez, J. Paradells: Wireless Home Automation Networks: A Survey of Architectures and Technologies, IEEE Communications Magazine, Vol. 48, No. 6, June 2010, pp. 92 – 101.

[8] S. Katsikeas, K. Fysarakis, A. Miaoudakis, A. Van Bemten, I. Askoxylakis, I. Papaefstathiou, A. Plemenos: Lightweight & Secure Industrial IoT Communications via the MQ Telemetry Transport Protocol, Proceedings of the IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, July 2017, pp. 1 – 8.

[9] J. Granjal, E. Monteiro, J. Sá Silva: Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues, IEEE Communications Surveys & Tutorials, Vol. 17, No. 3, Thirdquarter 2015, pp. 1294 – 1312.

[10] A. H. Alhamedi, V. Snasel, H. M. AIdosari, A. Abraham: Internet of Things Communication Reference Model, Proceedings of the 6th International Conference on Computational Aspects of Social Networks, Porto, Portugal, July 2014, pp. 61 – 66.

[11] G. Perrone, M. Vecchio, R. Pecori, 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, Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security, Porto, Portugal, April 2017, pp. 246 – 253.

[12] P. Fremantle, B. Aziz, J. Kopecky, P. Scott: Federated Identity and Access Management for the Internet of Things, Proceedings of the International Workshop on Secure Internet of Things, Wroclaw, Poland, September 2014, pp. 10 – 17.

[13] A. Fiat, A. Shamir: How to Prove yourself: Practical Solutions to Identification and Signature Problems, Proceedings of the Advances in Cryptology – Crypto'86, Santa Barbara, California, USA, August 1987, pp. 186 – 194.

[14] M. Schukat, P. Flood: Zero-Knowledge Proofs in M2M Communication, Proceedings of the 25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), Limerick, Ireland, June 2014, pp. 1 – 5.

[15] I.- H. Chuang, B.- J. Guo, J.- S. Tsai, Y.- H. Kuo: Multi-Graph Zero-Knowledge-Based Authentication System in Internet of Things, Proceedings of the IEEE International Conference on Communications (ICC), Paris, France, May 2017, pp. 1 – 7.

[16] V. V. Yashchenko: Introduction to Cryptography, 4th Edition, Publishing house of MCNMO, Moscow, 2012. (In Russian).

[17] M. Singh, M. A. Rajan, V. L. Shivraj, P. Balamuralidhar: Secure MQTT for Internet of Things (IoT), Proceedings of the 5th International Conference on Communication Systems and Network Technologies, Gwalior, India, April 2015, pp. 746 – 751.

[18] A. Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul, A. Panya: Authorization Mechanism for MQTT-Based Internet of Things, Proceedings of the IEEE International Conference on Communications Workshops (ICC), Kuala Lumpur, Malaysia, May 2016, pp. 290 – 295.

[19] J.- H. Han, J.- N. Kim: A Lightweight Authentication Mechanism Between IoT Devices, Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC), Jeju, South Korea, October 2017, pp. 1153 – 1155.

[20] A. Bhawiyuga, M. Data, A. Warda: Architectural Design of Token based Authentication of MQTT Protocol in Constrained IoT Device, Proceedings of the 11th International Conference on Telecommunication Systems Services and Applications (TSSA), Lombok, Indonesia, October 2017, pp. 1 – 4.

[21] K. M. Renuka, S. Kumari, D. Zhao, L. Li: Design of a Secure Password-Based Authentication Scheme for M2M Networks in IoT Enabled Cyber-Physical Systems, IEEE Access, Vol. 7, April 2019, pp. 51014 – 51027.

[22] A. K. Ranjan, M. Hussain: Terminal Authentication in M2M Communications in the Context of Internet of Things, Procedia Computer Science, Vol. 89, August 2016, pp. 34 – 42.

[23] A. Rabiah, K. K. Ramakrishnan, E. Liri, K. Kar: A Lightweight Authentication and Key Exchange Protocol for IoT, Proceedings of the Workshop on Decentralized IoT Security and Standards (DISS), San Diego, USA, February 2018, pp. 1 – 6.

[24] A. Esfahani, G. Mantas, R. Matischek, F. B. Saghezchi, J. Rodriguez, A. Bicaku, S. Maksuti, M. G. Tauber, C. Schmittner, J. Bastos: A Lightweight Authentication Mechanism for M2M Communications in Industrial IoT Environment, IEEE Internet of Things Journal, Vol. 6, No. 1, February 2019, pp. 288 – 296.

[25] L. Nastase: Security in the Internet of Things: A Survey on Application Layer Protocols, Proceedings of the 21st International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, May 2017, pp. 659 – 666.

[26] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, R. Al-Hatmi: Internet of Things: Survey and Open Issues of MQTT Protocol, Proceedings of the International Conference on Engineering & MIS (ICEMIS), Monastir, Tunisia, May 2017, pp. 1 – 6.

[27] W. Diffie, M. Hellman: New Directions in Cryptography, IEEE Transactions on Information Theory, Vol. 22, No. 6, November 1976, pp. 644 – 654.

[28] Standards for Efficient Cryptography, SEC 1: Elliptic Curve Cryptography, Version 2.0, May 2009.

[29] A. Venedyuhin: Keys, Ciphers, Messages: How TLS Works, Available at: https://tls.dxdt.ru/tls.html#ecdsa (accessed: 06.10.2019). (In Russian).

[30] B. Lum Jia Jun: Implementing Zero-Knowledge Authentication with Zero Knowledge (ZKA_wzk), Proceedings of the PyCon Asia-Pacific 2010 Conference, Singapore, Singapore, June 2010, pp. 1 – 19.

[31] B. Schneier, Applied Cryptography. Protocols, Algorithms and Source Code in C., 20th Edition, 2015.

[32] Moquette Java MQTT Lightweight Broker, Available at: https://github.com/andsel/moquette (accessed: 06.10.2019).