Message Queue Telemetry Transport

Subjects: Telecommunications

Contributor: Özlem Şeker , Gökhan Dalkılıç , Umut Can Çabuk

The Internet of things (IoT) accommodates lightweight sensor/actuator devices with limited resources; hence, more efficient methods for known challenges are sought after. Message queue telemetry transport (MQTT) is a publish/subscribe-based protocol that allows resource-efficient communication among clients, so-called brokers, and servers.

authorization HOTP IoT MQTT

1. Introduction

Many electronic devices containing sensors/actuators are integrated with networking capabilities to enable communications and enhanced automation. Many such devices can also establish Internet connections, resulting in the emergence of the Internet of things (IoT) concept. The devices that constitute the IoT may include sensors, actuators, interfaces, robotics, and storage. They gather, generate, store, and disseminate data and further exchange them with other devices connected to the internet. They may even manage themselves with no or little human interaction. Wireless sensor networks (WSNs) [1], the premise of the IoT, may include a large number of sensor nodes that are low-cost, small-sized, and have limited bandwidth, so they can communicate over specific lightweight protocols. Many modern IoT deployments include devices, functionalities, and goals similar to that of a WSN. This is why they also have similar constraints, although they support the Internet Protocol (IP) to some extent [2][3][4].

The number of devices connected to the Internet is dramatically increasing, thanks to the IoT and its industrial applications. Examples of current research topics include the relation between energy availability, sensing quality, and the overall throughput in cognitive radio networks powered by energy harvesting techniques [5] and maximizing packet collection performance in wireless-powered IoT networks [6]. Those highlight the significance of efficient energy management and data transmission strategies in IoT networks. On the other hand, IoT devices with limited bandwidth, memory, and computing capabilities are also vulnerable to security threats. Due to their limitations, traditional security mechanisms used in computers are unlikely to be implemented in IoT devices. In addition, the sensor nodes can reach out to the Internet via some gateways running on the application layer [7]. Therefore, the gateways need to be designed to interpret many different protocols. For the gateways and other intermediary elements, this poses many design challenges. Message queue telemetry transport (MQTT) (no longer accepted as an acronym since version 5.0) is a lightweight communication protocol widely adopted in IoT applications due to its resource efficiency, flexibility, and scalability [8]. However, it lacks advanced security measures that are highly desired in critical applications.

The constrained application protocol (CoAP), an alternative to MQTT, is also popular in IoT. It utilizes the user datagram protocol (UDP) [9] and is comparable to the hypertext transfer protocol (HTTP) since it is a web-based scheme. It provides reduced packet header sizes for resource-constraint devices. However, the CoAP header size is four bytes, whereas the MQTT header size is only two bytes [10]. Moreover, MQTT reportedly demonstrates better performance than CoAP in terms of packet losses and transmission latencies under increased network traffic [11]. Hence, MQTT is apparently more "lightweight", and thus, it is preferred over CoAP. Before proceeding to the details of the problem statement and the novel contributions, researchers provide some background regarding MQTT and other referred concepts.

1.1. Background

An introduction to the MQTT protocol and its security features is provided in this part. There is also a brief discussion on using a hash-based message authentication code (HMAC)-based one-time password (HOTP).

1.1.1. Overview of MQTT

The MQTT protocol was developed by IBM in 1999 and was standardized by the organization for the advancement of structured information standards (OASIS) in 2013 [12] to provide lightweight machine-type communications. Thanks to its publish/subscribe-based architecture, MQTT is a convenient communication protocol for most IoT scenarios. It is preferred for devices with limited resources [13][14][15].

Nodes in an MQTT-driven network may have different roles. The clients (i.e., mostly sensor/actuator devices) can either be publishers or subscribers for any information flow (e.g., sensed temperature data). Such flows of information are tagged with topic labels (called a "topic") considering their context (e.g., temperature). Hence, some clients may subscribe to certain topics and discard the rest.

There are also intermediary nodes, called brokers, responsible for distributing the information disseminated by the publishers to the interested subscribers. The broker keeps an up-to-date record of which client is interested in which topics. So, it shall forward the incoming data to the subscribers of that topic by filtering out the access rights. The publishers and subscribers do not have prior knowledge of each other. All the communication is carried out through the broker(s). A publisher and subscriber(s) do not have to be connected to the same network simultaneously [16]. Occasionally, a broker may also be a client at the same time [17]. Likewise, a publisher of one (or more) arbitrary topics may be a subscriber of others. Other types of nodes (e.g., servers, databases, etc.) may optionally co-exist in the ecosystem.

MQTT's message packets include three components: a mandatory 2-byte fixed-length header, an optional variable-length header, and an optional payload. The fixed-size header consists of the control field that contains the 4-bit message type field, the 4-bit control flag, and the packet length field. MQTT supports 14 unique message types that are connection request (*CONNECT*), connection acknowledgment (*CONNACK*), subscription request (*SUBSCRIBE*), subscription acknowledgment (*SUBACK*), unsubscription request (*UNSUBSCRIBE*), unsubscription acknowledgment (*UNSUBACK*), publish data (*PUBLISH*), publish acknowledgment (*PUBACK*), publish received (*PUBREC*), publish released (*PUBREL*), publish complete (*PUBCOMP*), ping request (*PINGREQ*), ping response (*PINGRESP*) and disconnection request (*DISCONNECT*).

1.1.2. Security Features in MQTT

Due to its lightness, there is no advanced security mechanism built into the MQTT protocol, except for a basic authentication scheme involving a username and password- control over a user database. When there are no resource-constrained devices in the network, the transport layer security (TLS) protocol can also be used with MQTT. It is the most secure way to provide data confidentiality and integrity [18]. However, this may not be applicable if lightweight devices are included, because allocating a secure end-to-end channel via TLS requires complex and heavy computations. If no security mechanism is implemented along with MQTT, then the subscribers may grant access to all the messages.

Moreover, a malicious publisher may act as an attacker and may cause a denial of service (DoS) attack. The use of access token(s) was already recommended to provide authentication and authorization mechanisms in the existing literature [19], but there is no viable security mechanism between the clients and an authorization server, except for the TLS-based hypertext transfer protocol secure (HTTPS). HTTPS allows using tokens to manage access control [16]. Furthermore, no prior studies have yet suggested a client-side authentication of the broker(s), which is necessary to achieve mutual trust and security within the ecosystem. This study provides a two-step mutual authentication and authorization mechanism using a “trusted” third-party authentication server that runs OAuth 2.0 [20]. It is also challenging to manage role-based access controls due to the potentially high numbers of connected devices. Since MQTT has no native security mechanism to address these issues, a dynamic token scheme that includes the scopes (i.e., authorized topics) of the clients is considered via the OAuth 2.0 protocol [21].

1.1.3. Using HOTP with MQTT

An authorization server (as discussed above), HOTP [22], and hash chains can be used to verify the identities of the client nodes and the broker(s) communicating via the MQTT protocol. HOTP already became a standard in 2005 [23] and is used to provide mutual authentication, as well as to prevent replay attacks without requiring time synchronization [24][25]. In the background, it relies on the HMAC, a unique secret (K), and also a counter value (C) that must be kept synchronized with the HOTP generator (and validator). A HOTP is generated through the truncation of a pre-computed HMAC-SHA-1 value, as shown below:

$$\text{HOTP}(K, C) = \text{Truncate}(\text{HMAC\_SHA-1}(K, C)) \quad (1)$$

The broker implements the HOTP generator role by combining the HOTP with Lamport’s backward hash chain scheme [22]. Nonetheless, since the exponentiation of such hash functions needs exhaustive computations on the clients (that are presumed to be constrained devices), a critical problem arises with Lamport’s original scheme. A feasible solution to that issue is to move the computationally heavy phases of the scheme to the brokers. Therefore, the hash chain is generated N-times by the broker itself, whereas the clients act as the HOTP validators and merely store the hash chain values [26]. The clients validate the broker while N decrements until reaching zero.

They have (HOTP (client ID, 3)) value; ‘3’ is considered due to the handshaking procedure. The broker generates the HOTP hash chain value N-times in total, as given below, where h is the hashing function

$$p = h\ (h\ (\ldots\ h\ (HOTP\ (clientID, 3))\ \ldots))\ (2)$$

Last but not least, the advanced encryption standard (AES) [27] (or equivalent) is required to provide data confidentiality in the links between the clients and the broker(s). In this way, subsistent data integrity can be offered, too.

2. Message Queue Telemetry Transport

The effectiveness of the related works is elaborated on, and unresolved issues are pointed out. Fremantle et al. [21] integrated the OAuth 2.0 protocol to provide means of access control to the MQTT communications. They claim that conventional authorization and authentication systems are not suitable for IoT scenarios because it is difficult to manage each device using a username–password pair and related role credentials, but they did not provide any user interface or interaction scheme between the machine(s) and the developer. The access token and the scope(s) are obtained by each client using the OAuth 2.0 protocol, that is, to manage the authentication and authorization mechanisms whenever a client connects to the MQTT broker. The MQTT protocol runs over the transmission control protocol (TCP), so it can utilize TLS to provide a secure channel in between, but most IoT devices are limited in terms of battery, storage, and computational power. Likewise, in their work, the connection between the clients and the MQTT broker is established without TLS due to using an 8-bit Arduino as the client device. Refreshing the token is challenging because of its limited storage. However, merely using the same token leads to security vulnerabilities such as eavesdropping, spoofing, and denial of service attacks. Moreover, exchanging the scope is also impossible since each scope is paired with an access token. Although it is an inspirational work, a viable solution must make use of dynamic tokens or passwords. Later, in 2015, Singh et al. [28] provided means of security to the MQTT protocol using lightweight elliptic curve cryptography (ECC) to encrypt the broadcast messages. The ECC was preferred to optimize the complexity. First, the publisher prepares an access list to send to the broker and demands an encryption key from the broker. It also adds some information about the user access rights to decrypt the messages and sends them to the broker. If a subscriber holds access permission for a specific message, it also gets the key from the broker and then decrypts the message with this private key. However, no information was given regarding how the key would have been generated. Furthermore, the publisher sends the access tree (a tree-formed data structure representing the access policy) to the broker and gets the key from the broker over a non-secure channel, so that an attacker may easily reach this information by eavesdropping.

In a home automation setup proposed by Upadhyay et al. [29], an access control list (ACL) plugin is added on top of the MQTT protocol to prevent access to sensor data. An ACL is a list file that includes usernames, encrypted passwords, and topic permissions in their model. This approach comes with two major consequences. First, in a dynamic and crowded ecosystem, the addition or removal of devices would be typical and continuous. This behavior naturally causes the addition or removal of new usernames, passwords, and access rights to the list.

Hence, the list has to be updated and re-shared repeatedly between the devices, which shall cause an undeniable overhead. Second, this approach only provides authorization but does not enforce any specific authentication control. An MQTT access control system was proposed by Niruntasukrat et al. [30]. It is based on OAuth 1.0 authorization protocol to avoid exhaustive computations. In addition, one more authentication factor was implemented to disseminate a piece of unique credential information over an HTTPS connection to secure the tokens. However, the security issues they addressed are only a portion of the possible risks, as they argued as well. Although providing authorization, the system lacks confidentiality aspects regarding the shared information. It is vulnerable to eavesdropping as well as replay, man-in-the-middle, and denial-of-service attacks.

Zamfir et al. [31] have made a comparative analysis of the security aspects of MQTT and CoAP protocols based on the pre-shared key approach and the use of raw public keys. The pre-shared key approach relies on symmetric encryption and the use of TLS between both ends of the communication. It can be used simply with CoAP but is not natively supported in MQTT. Such uses of TLS increase computational costs, too. The same limitations also apply to the raw public key method. Both protocols support certificates, but they require devastating computational power (and energy). However, payload encryption is made by AES. Yet, there is no extra computation or use of non-secure channels for key generation and distribution. Rajan et al. [32] proposed a mechanism where the broker uses a secure user profile-based data privacy by hierarchical inner product encryption (HIPE) for data subscription. So, they managed to allow access only to clients who are authorized in real-time video sharing. The study concludes that HIPE encryption and decryption lead to some delays in data transmissions. In addition, role-based access control that requires a tree-like hierarchy is not usually suitable for IoT scenarios.

The RES-256 algorithm introduced by Nagarajan et al. [33] offers an access control mechanism to enhance the security of healthcare systems, based on the so-called Internet of medical things, including advanced encryption mechanisms (e.g., RC6), secure key exchange protocols (e.g., ECDSA) and SHA256. Using RC6 for encryption, ECDSA for key exchange, and SHA256 for data integrity can contribute to securing sensitive data and ensuring the integrity of healthcare information as they are well-established cryptographic algorithms. However, the proposed mechanism does not mitigate the vulnerabilities associated with weak or short passwords due to relying on SHA256. Alshammari [34] proposed another method in the health domain, where the pre-shared public key is distributed to clients by the broker and the username, and the passwords are verified from the database by the broker. Fathy et al. [35] considered an IoT scenario centered on agricultural irrigation with sensors and actuator devices, where the authentication between clients and brokers needs to be ensured. They provided confidentiality solely between clients by utilizing the expeditious cipher (X-cipher) without emphasizing access authorization. The study also compared the runtime memory usage performance of the X-cipher, AES, and PRESENT encryption algorithms.

Shilpa et al. [36] proposed the use of secure reliable message communication (SEC-RMC) protocol along with Mosquitto to perform MQTT data transmission more securely. In the study, a public–private key exchange is made between Mosquitto and clients while it is encrypted via AES for the confidentiality of the data. The broker only verifies the topic information with the database and then shares it with the subscriber. Unlike this scheme, MARAS provides the right to access the topic and the client’s permission to publish or subscribe via OAuth 2.0 protocol. In relation to that, [37] compared the runtime performances of lightweight encryption algorithms in terms of encryption and decryption operations on MQTT payloads. The study evaluates AES-128 Galois/counter mode (GCM), GIFT-combined feedback (COFB), Romulus N1, and Tiny JAMBU. While payload encryption provides confidentiality for the data transfer, it does not address other crucial aspects of secure communication, including authentication, integrity, and protection against attacks like replay and impersonation. Relying solely on payload encryption is insufficient for comprehensive security in MQTT communication. MARAS approaches security holistically and addresses all aspects of secure communication to mitigate potential risks and vulnerabilities effectively.

When asymmetric encryption is preferred, ECC stands as another powerful alternative. Ref. [38] describes implementing a public-key encryption system based on ECC to secure data in an MQTT-based energy management system. Since ECC uses elliptic curves over finite fields to provide strong encryption with relatively smaller key sizes compared to other encryption algorithms, for future work it may be preferred for resource-constrained devices like Raspberry Pi in IoT applications. Patel et al. [39] also advocated the use of ECC. They proposed a new authentication approach for IoT applications, namely, a two-factor level-dependent authentication for generic IoT (LDA-2IoT), where they utilized elliptic curve cryptography. The proposed approach allows users at a specific level within the hierarchy to access sensor devices deployed either below or at the same hierarchical level. The study conducted a validation of the LDA algorithm within a communication environment based on MQTT over TLS as its secure channel, involving users, gateways, and sensor devices. Using ECC for encryption, Saqib et al. [40] introduced a framework to address significant security concerns in IoT-based networks. Their approach combines elliptical curve cryptography and hash chains to establish a signature-based 3-factor authentication system tailored for IoT environments. MARAS uses a dynamic session key for mutual authentication to prevent known session key attacks. Payload encryption and access authorization play a crucial role in ensuring the security of their MQTT communications. While the study may not have specifically highlighted these aspects, it is imperative to address payload encryption and access authorization when implementing a secure MQTT protocol.

A 2017 study by Katsikeas et al. [41] applied payload encryption using different AES modes to the data carried by the MQTT protocol. They have concluded that the AES-CBC is the best choice for constrained IoT devices. The study provides a solid knowledge base regarding AES-based encryption schemes (though it does not cover the AES-CFB mode). However, the research merely focuses on confidentiality and ignores other security aspects; yet, there is no explanation regarding the preferred key generation and distribution mechanisms. Without employing any authentication scheme, a malicious subscriber may capture all published information on any topic. Moreover, a malicious publisher can continuously flood spam messages and may eventually cause a DoS attack. Bhawiyuga et al. [42] have used a JavaScript object notation (JSON) web token (JWT) authentication server to authenticate the devices in the MQTT network. This was carried out via a direct token exchange between the clients and the broker; the broker then validates the tokens by sending them to the authentication server. However, although mentioned superficially, their work does not elucidate how a legitimate client device can maintain (or re-establish) its connection after its token is expired. Apparently, both the broker and the authentication server reject the connection requests in such a case. More importantly, while the token is a static one involving a hash algorithm to be protected against integrity attacks, it can still be obtained by eavesdropping. Hence, the token can be reused by malicious nodes repeatedly with a replay attack until it expires. To mitigate such vulnerabilities, a two-step authentication scheme involving a dynamic password mechanism must be implemented.

Bashir et al. [43] recommended using a trusted third-party server to handle key generation and distribution on MQTT. The server generates unique client IDs and distributes them to the clients. Thus, they are used as a pre-shared key for the encrypted transmission of a second key. The payloads of the clients are then encrypted with this second key. In their scenario, the initial distribution of the IDs will require a TLS connection unless they are sent as cleartext (which is worse). Further, this solution does not prevent replay attacks and flooding of vast messages by malicious nodes. So, attackers may easily overload the message traffic because the broker does not have a suitable device authentication mechanism. In their 2018 study, Wardana et al. [44] presented a prototype of an access control mechanism for IoT devices, which is based on the use of tokens along with the Redis authorization server. In addition, a secure socket layer (SSL) certificate was issued to provide data integrity and confidentiality between the cloud and the MQTT broker(s). The proposed mechanism is very concrete to its extent; nevertheless, it does not include a procedure for broker authentication (by the clients). Another novel security solution for MQTT was suggested by Calabretta et al. [45]. They offered a new approach for the MQTT protocol, based on the augmented password-authenticated key agreement (AugPAKE) algorithm over the ActiveMQ middleware. They also proposed a way to generate authorization tokens for each topic to provide topic-specific access control. Their study supersedes a similar prior work also based on the use of AugPAKE without TLS, published by Shin et al. [46], which lacks authorization features. Despite being broader, [45] ignores the authentication of the broker(s) by the clients, which is critical.

A paper written by Bali et al. [47] points out that a ciphertext-only attacker may also be able to listen to the traffic and decrypt the password right after the broker authenticates the first publisher and subscriber. They proposed a chaotic lightweight algorithm as a precaution against the mentioned ciphertext-only attacks. The algorithm ensures that the key is updated periodically by both the broker and the client(s). This steady work has two minor flaws: first, it uses TLS for the initial authentication; second, it stores a list of topics and authorized clients on the broker. Sundarrajan et al. [48] used advanced multiple encryption systems (AMES) based on the Hardy Wall encryption, requiring less bandwidth and minimal memory while utilizing the MQTT protocol. This is proposed to ensure the secure communication of messages without any packet loss. IoT device IDs and user IDs are also to be encrypted by the Hardy Wall encryption to provide authentication to some extent. However, their model presupposes an unjustifiable trust between the nodes. The paper also lacks information regarding security vulnerabilities, related attacks, and their potential effects, which are vital in such works. Public key cryptography is already a working solution to the authentication problem in MQTT, but key distribution is a major issue in this case. Aknın et al. [49] proposed an architecture that integrates blockchain and smart contracts into an MQTT setup to ensure data immutability and to automate authentication, publishing, and subscribing processes. In their architecture, MQTT protocol authentication relies on the use of one-time passwords (OTPs). To securely transmit the keys required for OTP generation and message encryption, TLS is utilized during the initial phase.

Sanjuan et al. [50], like [39], have proposed using smart cards to distribute keys to clients and brokers. They provided mutual authentication and payload encryption. Although reliable against most remote attacks, their method is vulnerable to physical attacks, like simply capturing the nodes and the smart cards. On the other hand, Amoretti et al. [51] have utilized a cloud server and a token-based authentication scheme. They also added support for multiple brokers via broker bridging. However, their scheme lacks mutual authentication and payload encryption. The issue of securing the MQTT protocol still remains up-to-date, as Blazy et al. in 2021 [52] proposed a comprehensive framework for securing the protocol beyond its basic encryption scheme. Their authentication mechanism relied on attribute-based signatures. While this framework could work well for many IoT systems, it might be too heavy for a network populated with lightweight sensors due to the quasi-complex computation requirements. In [17], the use of multiple MQTT brokers and the authentication of these MQTT brokers were suggested with JSON web tokens. Since the study does not cover security requirements, i.e., client-broker authentication, data confidentiality, and client authorization, it may be exposed to various attacks such as eavesdropping, man-in-the-middle attack, and impersonation attack.

References

1. Görmüş, S.; Aydın, H.; Ulutaş, G. Security for the Internet of Things: A survey of existing mechanisms, protocols and open research issues. J. Fac. Eng. Archit. Gazi Univ. 2018, 33, 1247–1272.
2. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. 2015, 17, 2347–2376.
3. Xu, L.D.; He, W.; Li, S. Internet of things in industries: A survey. IEEE Trans. Ind. Inform. 2014, 10, 2233–2243.
4. Chiang, M.; Zhang, T. Fog and IoT: An overview of research opportunities. IEEE Internet Things 2016, 3, 854–864.
5. Liu, X.; Xu, B.; Wang, X.; Zheng, K.; Chi, K.; Tian, X. Impacts of sensing energy and data availability on throughput of energy harvesting cognitive radio networks. IEEE Trans. Veh. Technol. 2023, 72, 747–759.
6. Song, X.; Chin, K.W. Maximizing Packets Collection in Wireless Powered IoT Networks with Charge-or-Data Time Slots. IEEE Trans. Cogn. Commun. Netw. 2023. accepted.
7. Zhu, Q.; Wang, R.C.; Chen, Q.; Liu, Y.; Qin, W.J. IoT Gateway: Bridging Wireless Sensor Networks into Internet of Things. In Proceedings of the IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing, Hong Kong, China, 11–13 December 2010; pp. 347–352.
8. Chien, H.-Y.; Wang, N.Z. A novel MQTT 5.0-based over-the-air updating architecture facilitating stronger security. Electronics 2022, 11, 3899.

9. Asim, M. A Survey on Application Layer Protocols for Internet of Things (IoT). Int. J. Adv. Res. Comput. 2017, 8, 996–1000.
10. Seoane, V.; Garcia-Rubio, C.; Almenares, F.; Campo, C. Performance evaluation of CoAP and MQTT with security support for IoT environments. Comput. Netw. 2021, 197, 108338.
11. Soni, D.; Makwana, A.A. Survey on MQTT: A Protocol of Internet of Things (IoT). In Proceedings of the International Conference on Telecommunication, Power Analysis and Computing Techniques (ICTPACT), Chennai, India, 6–8 March 2017; pp. 173–177.
12. OASIS MQTT Version 3.1.1 Plus Errata 01. Available online: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.pdf (accessed on 2 May 2023).
13. Thangavel, D.; Ma, X.; Valera, A.; Tan, H.; Tan, C.K. Performance Evaluation of MQTT and CoAP via a Common Middleware. In Proceedings of the IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 21–24 April 2014; pp. 1–6.
14. Naik, N. Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP. In Proceedings of the IEEE International Systems Engineering Symposium (ISSE), Vienna, Austria, 11–13 October 2017; pp. 1–7.
15. Yokotani, T.; Sasaki, Y. Comparison with HTTP and MQTT on Required Network Resources for IoT. In Proceedings of the International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, Indonesia, 13–15 September 2016; pp. 1–6.
16. Shinde, S.A.; Nimkar, P.A.; Singh, S.P.; Salpe, V.D.; Jadhav, Y.R. MQTT-message queuing telemetry transport protocol. Int. J. Res. 2016, 3, 240–244.
17. Azzedin, F.; Alhazmi, T. Secure data distribution architecture in IoT using MQTT. Appl. Sci. 2023, 13, 2515.
18. Munshi, A. Improved MQTT secure transmission flags in smart homes. Sensors 2022, 22, 2174.
19. Ragothaman, K.; Wang, Y.; Rimal, B.; Lawrence, M. Access control for IoT: A survey of existing research, dynamic policies and future directions. Sensors 2023, 23, 1805.
20. The OAuth 2.0 Authorization Framework. Available online: https://www.ietf.org/rfc/rfc6749.txt (accessed on 2 May 2023).
21. Fremantle, P.; Aziz, B.; Kopecky, J.; Scott, P. Federated Identity and Access Management for the Internet of Things. In Proceedings of the International Workshop on Secure Internet of Things, Wroclaw, Poland, 7–11 September 2014; pp. 10–17.
22. Park, C.S. One-time password based on hash chain without shared secret and re-registration. Comput. Secur. 2018, 75, 138–146.

23. HOTP: An HMAC-Based One-Time Password Algorithm. Available online: tools.ietf.org/pdf/rfc4226.pdf (accessed on 2 May 2023).

24. Kim, H.J.; Kim, H.S. AUTHHOTP-HOTP Based Authentication Scheme over Home Network Environment. In Computational Science and Its Applications—ICCSA 2011; Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B.O., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6784, pp. 622–637.

25. Saxena, A. Dynamic Authentication: Need than a Choice. In Proceedings of the 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE), Bangalore, India, 6–10 January 2008; pp. 214–218.

26. Yerlikaya, Ö. Security Enhanced Lightweight Messaging Protocol. Master's Thesis, Dept. Computer Engineering, the Graduate School of Natural and Applied Sciences, Dokuz Eylül University, İzmir, Turkey, 2018.

27. Blazhevski, D.; Stojcevska, B.; Pachovski, V. Modes of Operation of the AES Algorithm. In Proceedings of the 10th Conference for Informatics and Information Technology (CIIT 2013), Bitola, Macedonia, 18–21 April 2013; pp. 212–216.

28. Singh, M.; Rajan, M.A.; Shivraj, V.L.; Balamuralidhar, P. Secure MQTT for Internet of Things (IoT). In Proceedings of the 5th International Conference on Communication Systems and Network Technologies, Gwalior, India, 4–6 April 2015; pp. 746–751.

29. Upadhyay, Y.; Borole, A.; Dileepan, D. MQTT Based Secured Home Automation System. In Proceedings of the Symposium on Colossal Data Analysis and Networking (CDAN), Indore, India, 18–19 March 2016; pp. 1–4.

30. Niruntasukrat, A.; Issariyapat, C.; Pongpaibool, P.; Meesublak, K.; Aiumsupucgul, P.; Panya, A. Authorization Mechanism for MQTT-Based Internet of Things. In Proceedings of the IEEE International Conference on Communications Workshops, Kuala Lumpur, Malaysia, 23–27 May 2016; pp. 290–295.

31. Zamfir, S.; Balan, T.; Iliescu, I.; Sandu, F. A Security Analysis on Standard IoT Protocols. In Proceedings of the International Conference on Applied and Theoretical Electricity (ICATE), Craiova, Romania, 6–8 October 2016; pp. 1–6.

32. Rajan, M.A.; Varghese, A.; Narendra, N.; Singh, M.; Shivraj, V.L.; Chandra, G.; Balamuralidhar, P. Security and Privacy for Real Time Video Streaming Using Hierarchical Inner Product Encryption Based Publish-Subscribe Architecture. In Proceedings of the 30th International Conference on Advanced Information Networking and Applications Workshop, Crans-Montana, Switzerland, 23–25 March 2016; pp. 373–380.

33. Nagarajan, S.M.; Deverajan, G.G.; Chatterjee, P.; Alnumay, W.; Ghosh, U. Effective task scheduling algorithm with deep learning for Internet of health things (IoHT) in sustainable smart

cities. Sustain. Cities Soc. 2021, 71, 102945.

34. Alshammari, H.H. The Internet of things healthcare monitoring system based on MQTT protocol. Alex. Eng. J. 2023, 69, 275–287.
35. Fathy, C.; Ali, H.M. A secure IoT-based irrigation system for precision agriculture using the expeditious cipher. Sensors 2023, 23, 2091.
36. Shilpa, V.; Vidya, A.; Pattar, S. MQTT based secure transport layer communication for mutual authentication in IoT network. Glob. Transit. Proc. 2022, 3, 60–66.
37. Winarno, A.; Sari, R.F. A novel secure end-to-end IoT communication scheme using lightweight cryptography based on block bipher. Appl. Sci. 2022, 12, 8817.
38. Ramyasri, G.; Murthy, G.R.; Itapu, S.; Krishna, S.M. Data transmission using secure hybrid techniques for smart energy metering devices. e-Prime-Adv. Electr. Eng. Electron. Energy 2023, 4, 100134.
39. Patel, C.; Doshi, N. LDA-2IoT: A level dependent authentication using two factor for IoT paradigm. Comput. Netw. 2023, 223, 109580.
40. Saqib, M.; Jasra, B.; Moon, A.H. A lightweight three factor authentication framework for IoT based critical applications. J. King Saud Univ. 2022, 34, 6925–6937.
41. Katsikeas, S.; Fysarakis, K.; Miaoudakis, A.; Bemten, A.V.; Askoxylakis, I.; Papaefsta-thiou, I.; Plemenos, A. Lightweight & Secure Industrial IoT Communications via the MQ Telemetry Transport Protocol. In Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, 3–6 July 2017; pp. 1193–1200.
42. Bhawiyuga, A.; Data, M.; Warda, A. Architectural Design of Token Based Authentication of MQTT Protocol in Constrained IoT Device. In Proceedings of the 11th International Conference on Telecommunication Systems Services and Applications (TSSA), Lombok, Indonesia, 26–27 October 2017; pp. 1–4.
43. Bashir, A.; Mir, A.H. Securing communication in MQTT enabled Internet of things with lightweight security protocol. EAI Endorsed Trans. Internet Things 2017, 3, e1.
44. Wardana, A.A.; Perdana, R.S. Access Control on Internet of Things Based on Publish/Subscribe Using Authentication Server and Secure Protocol. In Proceedings of the 10th International Conference on Information Technology and Electrical Engineering (ICITEE), Bali, Indonesia, 24–26 July 2018; pp. 118–123.
45. Calabretta, M.; Pecori, R.; Velti, L. A Token-Based Protocol for Securing MQTT Communications. In Proceedings of the 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 13–15 September 2018; pp. 1–6.

46. Shin, S.; Kobara, K.; Chuang, C.C.; Huang, W. A Security Framework for MQTT. In Proceedings of the IEEE Conference on Communications and Network Security (CNS), Philadelphia, PA, USA, 17–19 October 2016; pp. 432–436.
47. Bali, R.S.; Jaafar, F.; Zavarasky, P. Lightweight Authentication for MQTT to Improve the Security of IoT Communication. In Proceedings of the ACM International Conference Proceeding Series, Kuala Lumpur, Malaysia, 19–21 January 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 6–12.
48. Sundarrajan, M.; Narayanan, A.E.A. An authentication scheme for an IoT environment using advanced multiple encryption system. Int. J. Innov. Tech. Expl. Eng. 2019, 9, 868–874.
49. Aknin, R.; Bentaleb, Y. Enhanced MQTT architecture for smart supply chain. Int. J. Adv. Comput. Sci. Appl. 2023, 14, 861–869.
50. Sanjuan, E.B.; Cardiel, I.A.; Cerrada, J.A.; Cerrada, C. Message queuing telemetry transport (MQTT) security: A cryptographic smart card approach. IEEE Access 2020, 8, 115051–115062.
51. Amoretti, M.; Pecori, R.; Protskaya, Y.; Veltri, L.; Zanichelli, F. A scalable and secure publish/subscribe-based framework for industrial IoT. IEEE Trans. Ind. Inform. 2021, 17, 3815–3825.
52. Blazy, O.; Conchon, E.; Klingler, M.; Sauveron, D. An IoT attribute-based security framework for topic-based publish/subscribe system. IEEE Access 2021, 9, 19066–19077.

Retrieved from https://encyclopedia.pub/entry/history/show/110671