International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 5, October 2021, pp. 4577~4586
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i5.pp4577-4586

MQTT-PRESENT: Approach to secure internet of things applications using MQTT protocol

Imane Sahmi[1], Abderrahim Abdellaoui[2], Tomader Mazri[3], Nabil Hmina[4]
[1,4]Systems Engineering Laboratory, Sultan Moulay Slimane University, Beni Mellal, Morocco
[2]Systems Engineering Laboratory, Ibn Tofail University, Kenitra, Morocoo
[3]Electrical Systems and Telecommunications Engineering Laboratory, Ibn Tofail University, Kenitra, Morocco

Article Info

Article history:

Received Oct 28, 2020
Revised Apr 11, 2021
Accepted Apr 22, 2021

Keywords:

AugPAKE algorithm
IoT
MQTT
PRESENT
Security

ABSTRACT

The big challenge to raise for deploying the application's domain of the Internet of Things is security. As one of the popular messaging protocols in the IoT world, the message queue telemetry transport (MQTT) is designed for constrained devices and machine-to-machine communications, based on the publish-subscribe model, it offers a basic authentication using username and password. However, this authentication method might have a problem in terms of security and scalability. In this paper, we provide an analysis of the current research in the literature related to the security for the MQTT protocol, before we give a brief description of each algorithm used on our approach, to finally propose a new approach to secure this protocol based on AugPAKE algorithm and PRESENT encryption. This solution provides mutual authentication between the broker and their clients (publishers and subscribers), the confidentiality of the published message is protected twice, the integrity and non-repudiation of MQTT messages which is protected during the process of transmission.

Corresponding Author:

Sahmi Imane
Systems Engineering Laboratory
Sultan Moulay Slimane University
Av Med 5, P.O. Box 591,Pc 23000, Beni Mellal, Morocco
Email: sahmi.imane@univ-ibntofail.ac.ma

1. INTRODUCTION

The internet of things (IoT) plays an important role in our daily life. Cisco predicted at the end of 2020 that the number of connected devices will rise to 50 billion [1], the main objective of IoT is to form a network of identifiable objects like the radio frequency identification (RFID) connected to the Internet and can communicate with each other anytime, anywhere, and everywhere [2]. The IoT is the backbone of each application's domain like healthcare, agriculture, transportation, cities in general. The researchers are under pressure to quickly deploy this technology but security is a critical point without we can trust any application because of the big amount of sensitive data transmitted by devices [3]. A lot of works are made to find a solution to secure the platform used in the IoT application's domain. There are various IoT application layer protocols like advanced message queuing protocol (AMQP), constrained application layer protocol (CoAP), message queue telemetry transport (MQTT). A lot of advantages make the MQTT protocol the most used protocol in IoT applications [4], [5] due to its simplicity and lightweight. There are some security features to make into consideration in an IoT environment like confidentiality, integrity, and availability. It's a big challenge for researchers to propose solutions that verify these features adapted to constrained devices. In this perspective, we propose an approach to secure the MQTT protocol, a solution based on AugPAKE algorithm

and PRESENT encryption, that provides mutual authentication between the broker and their clients (publisher/subscriber), the confidentiality of the published message, the integrity, and non-repudiation of MQTT messages which is protected during the process of transmission. The paper has the following structure in section 2 we review the main contributions in the literature regarding the security solutions proposed for MQTT based on the authentication, the authorization, and cryptographic mechanism, and AugPAKE algorithm similar to our solution, in section 3 we detail the theoretical basis used in our approach (MQTT protocol and AugPAKE algorithm, PRESENT Encryption), while in section 4 we describe the proposed solution based on AugPAKE algorithm, the different steps in detail, and the exchanged messages. Finally, to explain the main security features approved by our contribution and some future works.

2. LITERATURE REVIEW

Security is a crucial point in the IoT domain, a lot of researchers are working to propose a solution for securing applications based on a constrained device. The big challenge is to apply security features such as authentication and authorization to verify the confidentiality, integrity, and availability of data information. Some authors propose a solution based on authentication like the authors Bali [6], propose a lightweight authentication mechanism based on a chaotic algorithm using self-key agreement and block cipher to improve the MQTT security, after simulation, it was mentioned that the data confidentiality was proven. Bhawiyuga [7], proposes an implementation of token-based authentication of MQTT protocol in constrained devices, but the proposed design consists of four components: Publisher, subscriber, MQTT broker, and Token authentication server. Rahman [8] the authors propose a secure version designed for wireless sensor networks by developing a multi-tier authentication system to ensure data privacy in IoT system depending on the use of ciphertext-policy attribute-based encryption (CP-ABE) or key-policy attribute-based encryption (KP-ABE) with the lightweight elliptic curve cryptography (ECC).

Other works are based on authorization mechanism to secure the protocol MQTT as: Niruntasukrat [9] their solution is based on O.Auth 1.0a an open authorization standard for web applications, the proposed system is based on a set of credentials (device ID, device secret, access token, and access token secret), some of these credentials need to be sent through a device which has to be not constrained and can afford HTTPS.

Other authors propose a cryptographic algorithm to secure MQTT protocol like in [10] they propose a composite security solution for MQTT by combining on Attribute-based encryption and advanced encryption standard (AES) S-boxes. The solution is based on public-key cryptography and secret-key cryptography, so double decryption must be done by the subscriber. We will have the same drawback more overhead. Another solution based on attributes is [11] Singh propose a new secure publish command "SPublish" for (MQTT, MQTT-SN) which encrypt data based on a set of attribute-based encryption (ABE), the key policy KP-ABE and a ciphertext policy CP-ABE using lightweight ECC, the use of CP-ABE causes an extra overhead which is the main drawback of using ABE. Mektoubi [12] their contribution is to secure the flow of distributed messages between the users of the MQTT protocol by using a certification authority that generates a private key and a certificate to a topic that is published for certified clients. The contribution in [13] proposes to secure MQTT by using access control lists (ACLs) embedded in a Mosquitto broker which acts like a filter allowing only those data which are requested thereby saving the flow of ambiguous data. The (ACL) method act as extra security to the whole process, for each data username and password, are created to get access to the data. However, this solution can cause more overhead, scaling with the number of different data to be transmitted. Two works are like our solution based on augmented password-only authentication and key exchange (AugPAKE) are: Shin [14], the solution called AugMQTT: A session key will be established between the publisher and the broker and another session between the broker and the subscriber without certifications. To publish a message, the publisher uses a secure symmetric-key encryption schema or a secure authenticated encryption with associated data schema to send data to the broker, the broker uses the same key to decrypt data and store the data message, for necessary request the broker transmits a ciphertext to the subscriber using the same key shared with the broker. The proposed solution is implemented through Mosquitto. Calabretta [15] MQTT-Auth the authors use the AugPAKE protocol to make the session key and use an authentication token wich create and publish on a certain topic and an authorization tokenshared between the publisher and somechosen subscribers. The solution uses ActiveMQ. This Table 1 makes a comparison between some proposed solutions to secure MQTT by giving some advantages and drawbacks for each one.

Table 1. Review of some proposed solution to secure MQTT

Ref	Proposed solution	Methods used	Advantages	Drawbacks
[6]	A lightweight authentication mechanism	Chaotic algorithm and block cipher	Data confidentiality	Not designed for multiple clients.
[7]	A token-based authentication	Json Web Token (JWT) authentication server.	Perform the authentication of the valid and expired token	Need a third party: a token authentication server.
[11]	A secure version of lightweight MQTT	Ciphertext-policy-based (KP/CP-ABE) lightweight (ECC)	Data privacy. secure under CPA and CCA, a man in the middle, and collision attack.	New message 'SPublish' not yet deployed on real IoT platform.
[9]	An authorization mechanism for MQTT security	OAuth 1.0a and a set of credentials	Protection against: eavesdropping attacks MITM attack, replay attack, node capturing the attack	A third party: AuthServer is needed.
[10]	A composite security framework for MQTT	Public-key cryptography and secret-key cryptography	Provides both confidentiality and fine-grained access control of data	More overhead.
[12]	Secure the flow of distributed message between the users	Certification authority private key	The solution displays an acceptable level of maturity for practical cases	Network saturation in case of a significant number of clients.
[13]	Using an Access Control Lists	ACL method	Broker as a filter, Saving the flow of ambiguous data	More overhead with the different messages
[16]	Modern fuzzing based MQTT	Fuzzer, sniffer, Mitmfuzzer	Focuses on testing	Does not propose security features
[17]	A lightweight authentication mechanism	Based on hash and XOR operations preshared keys between sensors and servers	Low computational cost, communication, and storage overhead, against the following attacks: replay, man-in-the-middle, impersonation, and modification	Designed for IoT environment in general The application's protocol does not define.
[14]	AugMQTT	AugPAKE algorithm	An authentication, confidentiality, and integrity of MQTT messages.	The broker isn't protected, the message is encrypted on code ASCII.
[15]	MQTT-Auth	AugPAKE algorithm, an authentication token, and an authorization token	Guaranteeing confidentiality authorization in accessing a topic.	A secondary channel is needed for tokens
[18]	A solution that supports 3 security levels for MQTT	Elliptic Curve Integrated Encryption Scheme	Secure message publishing of non-sensitive but not-modified data, secure data publishing from authenticated publishers to authenticated subscribers.	The solution depends on which security level for MQTT is wanted.

3. THE COMPREHENSIVE THEORETICAL BASIS

In our approach, we use the AugPAKE algorithm and the PRESENT encryption through the MQTT protocol, so we have to explain in detail the basis of each one. To finally propose the new approach of our solution to secure the MQTT protocol.

3.1. The message queuing telemetry transport protocol (MQTT)

The MQTT protocol is one of the IoT's application protocols. This is the most widely adopted protocol for developing IoT applications compared with the others application protocols [19]. MQTT was developed by Andy Stanford-Clark of IBM [20] and Arlen Nipper of Arcom in 1999 and standardized in 2013 by the Organization for the advancement of structured information standards (OASIS) [21]. It's suitable for IoT applications due to its simple model and low bandwidth usage. Furthermore, it guarantees the reliability of packet delivery. Some features of the MQTT protocol: It's a published/subscribe protocol and runs over TCP/IP. It's suited for constrained environments due to its simplicity and open source code. It also supports three levels of quality of service (QoS).

In MQTT, the publish-subscribe model mentioned below in Figure 1, the publisher sends the data to the broker for publishing in the 'Publish' message; a subscriber authenticates and subscribes to the broker for

a certain topic on the 'Subscribe' message, the broker sends the data to the specifics subscribers that are subscribed to the specific topic on 'Publish' message.

3.2. The AugPAKE protocol

The AugPAKE protocol [18] is based on the modification of the Diffie Hellman key exchange protocol. In this protocol, the client computes a key from the password to authenticate to the server which does not need to store client passwords. After successful authentication, the secure session is established and key agreement between the client and the server. The AugPAKE protocol exchanges 4 messages. The Augmented is a client-server session key establishment protocol that uses some parameters which are explained in this Table 2.

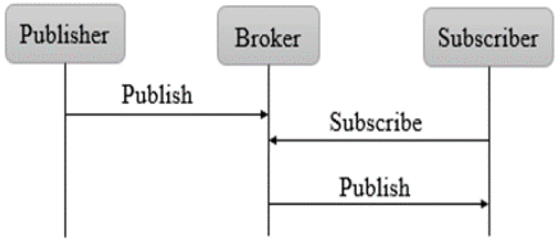

Figure 1. Publish-subscribe model

Table 2. Description of different variables

Variable	Description
G	The cyclic group of order q
g	Generator of G
p	A prime number such as p=aq+1, with an integer
H	Hash function returning a binary string of length k
H'	Hash function returning an integer
C	Client identifier
S	Server identifier
$\|\|$	Juxtaposition or concatenation of strings

The AugPAKE algorithm consists of two-phase: the initialization phase and the actual protocol execution:

The AugPAKE Algorithm

inputs: C: client identifier w: client's password S: Server identifier

initialization:

client C computes $W = g^{w'} \bmod p$ (1)

where $w' = H(C\|S\|w)$

client C ***send (C, W) to server S***

client C chooses x and computes $X = g^x \bmod p$ (2)

client C ***send (C,X) to server S***

traitement:

if server **S** received X equal to 0,1 or -1 (mod p)

server S discards the session

if not

server S chooses y and computes $Y = (XW^r)^y mod p$ (3)

where $r = H'(''01''\|C \| S \| X)$

server S send (S, Y) to the client C

if **client C** received Y equal to 0,1 or -1 (mod p)

client C terminates the protocol execution

If not

client C computes $K = Y^z mod\ p$ (4)

where $z = \frac{1}{x+w'r} mod\ q$

$r = H'(''01'', C, S, X)$

and $Vc = H(''02''\|C\|S\|X\|Y\|K)$ (5)

client C sent Vc to server S

if **server S** receives

$Vc \neq H(''02''\|C\|S\|X\|Y\|K)$

with K= $g^y mod\ p$ (6)

server S stops the procedure

if not

server S generates an authenticator

$Vs = H("03"\|C\|S\|X\|Y\|K)$ (7)

server S sends Vs to client C

server S computes the session key

$Sk = H(''04''\|C\|S\|X\|Y\|K)$ (8)

if **client C** receives

Vs $\neq H("03"\|C\|S\|X\|Y\|K)$

client C terminates the procedure

if not

Client C computes

$Sk = H(''04''\|C\|S\|X\|Y\|K)$

3.3. PRESENT

After secure key session establishment, we will use the payload encryption encapsulated into PUBLISH message. Encryption like a technique of cryptography which is the science of protecting data from malicious acts to ensure the confidentiality, integrity, and authenticity of the transmitted information. In this perspective, using the cryptographic technique is an intelligent choice to be developed to support heterogeneity, interoperability, key size, low energy consumption, limited resources of IoT devices [22]. That's why, we use PRESENT [23] encryption as a lightweight block cipher, compared to AES, it's 2,5 smaller than the last one. It's an example of the SP-Network cryptographic algorithm. It consists of 32 rounds, the block length is 64 bits and two supported key lengths of 80 and 128 bits as shown in Figure 2, the procedure of encryption is divided into three phases:

a. Key addition: the data undergoes a xor operation with the key from which only the last 64 bits of the result are taken for this operation. After this, the 64 bits data is divided into 16 blocks each containing 16 bits for the next process.
b. Non-linear substitution layer: Each of this block is then passed through the substitution boxes where the value in these boxes are replaced by the values in the substitution blocks.
c. Bit-wise permutation: The substituted data is then passed through the permutation block where all the 64 bits of the input data are reorganized. After this, the key is updated to produce another round key which is used next time the same entire process is repeated for the same set of data. This is repeated 32 times and then the encrypted data is transmitted.
d. PRESENT Algorithm

generateRoundKeys()
for i=1 to 31 **do**
addRoundKey(STATE;K_i)
sBoxLayer(STATE)
pLayer(STATE)
end for
addRoundKey(STATE,K_{32})

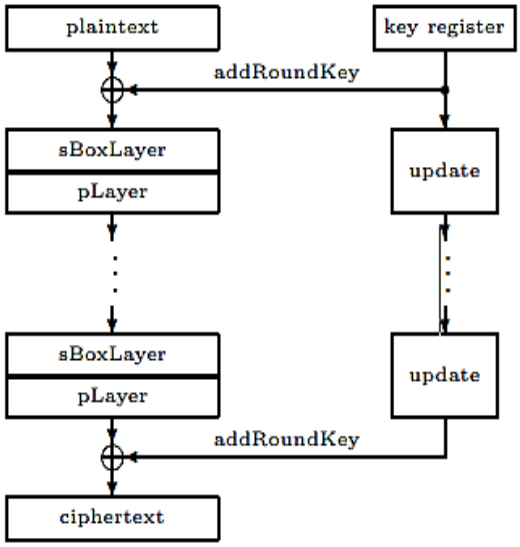

Figure 2. Algorithm description of PRESENT [23]

4. THE PROPOSED APPROACH

Our approach to secure the MQTT protocol consists of using the AugPAKE algorithm with PRESENT as lightweight encryption for the payload of PUBLISH message. Our contribution related to Aug-MQTT is to bring more lightness to the process of securing MQTT communication. In Aug-MQTT the authors use only secure symmetric-key encryption as AES which is heavy for constrained resources also they use a client to broker encryption but in our contribution we use an end to end encryption which the broker has no idea about the payload, it's encrypted by PRESENT and will be sent to the subscriber securely. In Aug-MQTT, the broker is a good target for a hacker, the data isn't secure, the message is plaintext and also in

MQTT-Auth, the authors mentioned that the broker could be a single point of attack. It has complete visibility of the exchanged data because it's in charge of encrypting and decrypting the MQTT payloads.

The diagram above present our approach based on AugPAKE and PRESENT. The first step is based on the AugPAKE algorithm to establish a secure key session between the publisher/the subscriber and the broker. It's divided into two phases:

a. The initialization phase: the publisher sends (Cp, Wp) to the broker where Cp: the identifier of the publisher and computed Wp by using the (1) in the AugPAKE algorithm:
b. The execution phase <Publisher -- Broker>: composed of multiple steps mentioned below in Figure 3, that explain the Key session between the Publisher and the Broker:

1) The publisher sends (C_p,X_p) to the broker according to (2).
2) The broker sends (S, Y_p) to the publisher when it received the right X_p; according to (3).
3) The publisher sends (Cp,Vcp); according to (5); when it received the right Yp.
4) The broker computes SK_p according to (8) and sends (S, V_{sp}), according to (7).
5) Finally, the publisher computes SKP; according to (8), when it's received the right Vsp. The publisher and the broker share a secure session key SKp.

c. The execution phase <Broker -- Subscriber>: composed of multiple steps mentioned below in Figure 4, that explain the Key session between the Broker and the Subscriber:

1) The Subscriber sends (C_s,X_s) to the broker according to (2).
2) The broker sends (S, Y_s) to the Subscriber when it received the right X_s; according to (3).
3) The Subscriber sends (C_s,V_{cs}); according to (5); when it received the right Y_s.
4) The broker computes SKs according to (8) and sends (S, Vss), according to (7).
5) Finally, the Subscriber computes SK_s; according to (8), when it's received the right Vss. The Subscriber and the broker share secure session key Sk_s.

The broker has the two sessions keys shared by the subscriber and the publisher as are mentioned in Figure 5, It computes the key K_s by the (9):

$$ks = Skp \oplus Sks \tag{9}$$

The key K_s must be shared with the publisher and the subscriber that's why the broker sends the key to the publisher (subscriber) by PRESENT encryption using the Skp (Sks). Both the publisher and the subscriber have the shared key K_s. When the publisher wants to publish a message, the PRESENT encryption process applied to the payload of the message PUBLISH using K_s, the publisher will transmit E(Ks, *Data1*) to the broker. When necessary/requested, Broker transmits a ciphertext E(K_S, *Data1*) to Subscriber, who can decrypt by PRESENT decryption with the key K_S to get the message published according to its corresponding topic.

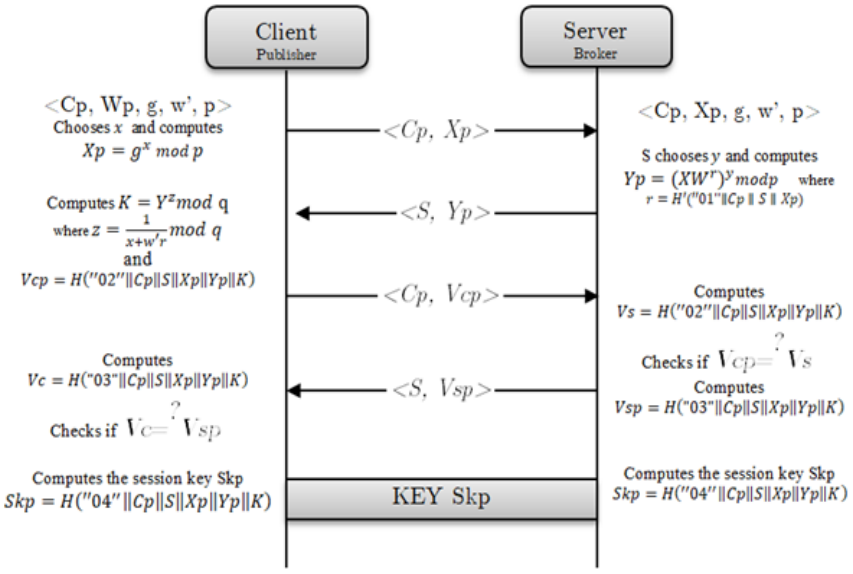

Figure 3. Key session between publisher/broker

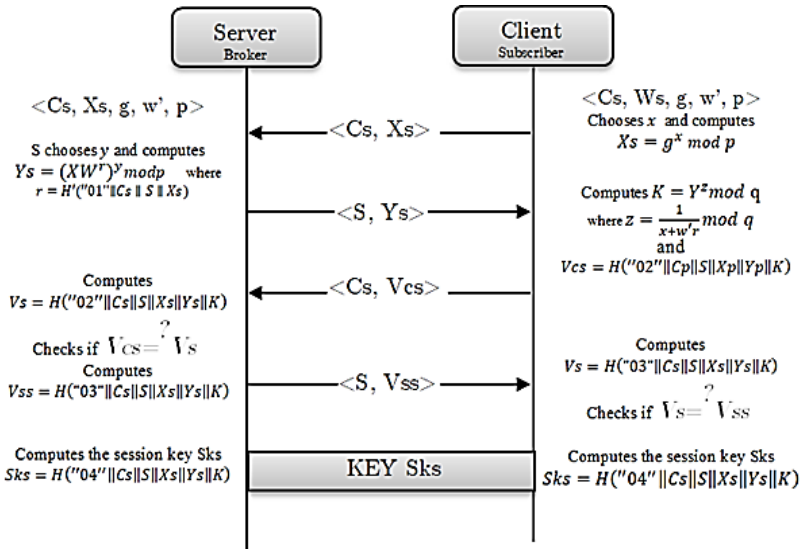

Figure 4. Key session between broker/subscriber

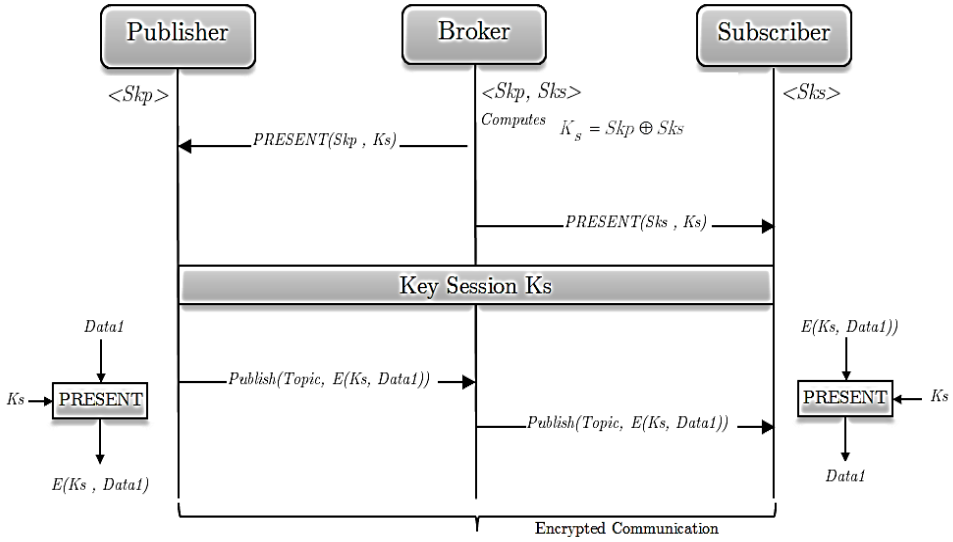

Figure 5. Encryption/decryption between publisher/subscriber

5. SECURITY DISCUSSION

In this section, we discuss the security properties that are supported by our solution:

a. Authentication: Everything on the Internet of Things must be identifiable and authenticated with the other objects connected, so it's a challenging task because of the nature of IoT. Our proposed solution provides mutual authentication between the broker and their clients (publishers and subscribers) [24].

1) Publisher (subscriber) authentication: the publisher (the subscriber) is authenticated by using the session key Sk_p (Sk_s) approved by the usage of the AugPAKE Algorithm.
2) Broker authentication: only the broker with his common key (Sk_p) or (Sk_s) can decrypt the publish and subscribe messages from publishers and subscribers who use the common broker keys.

b. Confidentiality: It is very important to ensure that the data is secure and only available to authorized users. Another point that the users of IoT must be aware of how the data is managed and ensure that the data is protected through the process [25]. In the proposed solution the published message is protected twice, first when it is transferred to the broker, by using the secure session generated by the AugPAKE Algorithm; only the client who has the session key can decrypt the message and in the second time in the side of the broker doesn't have the message in plaintext due to the PRESENT encryption.

c. Integrity and non-repudiation: It's very important to ensure the accuracy of the data; that it is created or modified by the authorized party only. The integrity feature can be imposed by maintaining end-to-end security in IoT communication. This is what our solution providers, the end-to-end security, and not the client to end security The data is encrypted by the PRESENT encryption during the process of publishing the message, the broker has no idea about the message transmitted [24].

Our contribution consists of reinforcing the security of the broker, the data is stored by the broker in an encrypted format (PRESENT encryption). The proposed system provides the authentication of publisher and subscriber: they are authenticated with the broker by using the session key Sk_p (Sk_s) approved by the usage of the AugPAKE algorithm. The confidentiality, integrity, and non-repudiation of MQTT messages from the publisher to the subscriber because the broker does not have the message in plaintext due to the PRESENT encryption.

By using the AugPAKE protocol in our approach, an attacker can get (C_p,X_p) ,(S,Y_p),V_{sp} or V_{cp} by eavesdropping, but he cannot compute an authenticated session key (Sk_p) shared between the publisher and the broker, therefore it's secure against passive attacks and also if the attacker controls the exchange message, he can't compute an authenticated session key so it's secure against active attacks (man in the middle attack/replay attack) also for the offline/online dictionary attacks [26]. Our contribution related to the previous works using the AugPAKE Protocol that we add the PRESENT encryption that provides more protection to the broker, the message will be encrypted. Our solution provides:

a. User anonymity: For the clients (subscriber/publisher) the identity of the clients is protected using Sk_s/Sk_p.
b. Mutual authentication: By using the session channel Sk_s (Sk_p) between the subscriber/Broker and publisher/Broker. The proposition validates important points of security for IoT using the MQTT protocol. The proposed solution is secure against:
c. Man in the middle attack: The attacker interfere between two nodes by monitoring, eavesdropping, and controlling the communication between the two sensor nodes to access the restricted data [27],
d. The replay attack and the offline password guessing attack: even if the attacker can guess a password, the parameter K in the (4) of the AugPAKE Algorithm is derived independently from the guessed password.

We make this Table 3 which summarize some advantages of our proposed solution compared with some previous works, some of these advantages: User anonymity, mutual authentication, data confidentiality data privacy, data integrity, and robustness against some attacks like Man in the middle, the offline password guessing attack, the replay attack. The results seem interesting in comparison with the other solutions working on the security of MQTT.

The second important parameter to take into consideration is the estimated cost time for this solution, according to the number of hash, modular exponentiation, and encryption operation, we do a comparison of the estimated cost time of each solution using AugPAKE (Aug-MQTT, Auth-MQTT, Our Approach) by making these parameters, the result is summarized in Table 4. The results of our solution in terms of the parameters cited in Table 4, look promising in comparison with the other solutions, we are on the way to validate these results by simulation.

Table 3. Some advantages of the proposed solution

	UA	*MA*	*MITM*	*OFP attack*	*Replay attack*	*DC*	*DP*	*DI*	*A*
[6]	×	✓	×	×	×	✓	×	×	×
[11]	×	×	✓	×	×	×	✓	×	×
[9]	×	×	✓	×	✓	×	×	×	×
[17]	×	×	✓	×	✓	×	×	×	×
[14]	×	✓	×	×	×	✓	×	✓	✓
[15]	✓	✓	✓	✓	✓	✓	×	×	✓
Our Approach	✓	✓	✓	✓	✓	✓	✓	✓	×

Note: UA : User Anonymity
MA:Mutual authentication
MITM: Man in the middle
OPG:Offline password guessing attack
DC: Data confidentiality
DP:Data privacy
DI: Data integrity
A:Authorization

Table 4. Comparison of the estimated cost time

Solution	Estimated cost Time of each solution
Aug-MQTT	$6\ T_h+4\ T_{ex}+4\ T_{ed}$
Auth-MQTT	$6\ T_h+6T_{ex}+4\ T_{ed}$
Our Approach	$6\ T_h+4\ T_{ex}+6\ T_{ed}$

Note: T_h = hash time; T_{ex}= Modular exponentiation time; T_{ed} = encryption/decryption time

6. CONCLUSION

Our contribution consists in this paper, to propose a new approach that makes the MQTT protocol, the most used protocol in IoT applications, more secure and competitive with the other solutions proposed by the others research to secure the MQTT protocol. Our solution is based on AugPAKE Algorithm and PRESENT lightweight encryption. Our contribution consists of using the PRESENT encryption comparing with works using the same AugPAKE algorithm the Aug-MQTT and MQTT- Auth, that makes the solution lightweight and more secure, the solution provides the user anonymity, mutual authentication, data confidentiality, and data privacy, data integrity and *non-repudiation* of data information also it's secure against some attacks like the man in the middle, offline password guessing, the replay attack. Finally, the estimated cost time of our proposed solution seems interesting to go ahead. For future works, we are on the way to simulate our approach in the publish/subscribe model and especially for the smart healthcare application. In others words, we will simulate some attacks to evaluate the robustness of our approach. Also, we are working on how Machine learning techniques can help us to secure the publish/subscribe model.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw*, vol. 54, no. 15, Oct. 2010, doi: 10.1016/j.comnet.2010.05.010.

[2] I. Yaqoob *et al.*, "Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges," *IEEE Wirel. Commun.*, vol. 24, no. 3, pp. 10–16, Jun. 2017, doi: 10.1109/MWC.2017.1600421.

[3] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: A survey," *J. Netw. Comput. Appl.*, vol. 88, pp. 10–28, 2017, doi: 10.1016/j.jnca.2017.04.002.

[4] M. B. Yassein, M. Q. Shatnawi, and D. Al-zoubi, "Application layer protocols for the Internet of Things: A survey," in *2016 International Conference on Engineering & MIS (ICEMIS)*, Agadir, Morocco, Sep. 2016, pp. 1–4, doi: 10.1109/ICEMIS.2016.7745303.

[5] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017, doi: 10.1109/JIOT.2017.2683200.

[6] R. S. Bali, F. Jaafar, and P. Zavarasky, "Lightweight authentication for MQTT to improve the security of IoT communication," In *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy (ICCSP '19)*, Association for Computing Machinery, New York, NY, USA, 2019, pp. 6-12, doi: 10.1145/3309074.3309081

[7] A. Bhawiyuga, M. Data, and A. Warda, "Architectural design of token based authentication of MQTT protocol in constrained IoT device," in *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*, 2017, pp. 1-4, doi: 10.1109/TSSA.2017.8272933.

[8] A. Rahman, S. Roy, M. S. Kaiser, and M. S. Islam, "A Lightweight Multi-tier S-MQTT Framework to Secure Communication between low-end IoT Nodes," in *2018 5th International Conference on Networking, Systems and Security (NSysS)*, 2018, pp. 1-6, doi: 10.1109/NSysS.2018.8631379.

[9] A. Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul, and A. Panya, "Authorization mechanism for mqtt-based internet of things," in *2016 IEEE International Conference on Communications Workshops (ICC)*, 2016, pp. 290-295, doi: 10.1109/ICCW.2016.7503802.

[10] L. Bisne and M. Parmar, "Composite secure MQTT for Internet of Things using ABE and dynamic S-box AES," in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, 2017, pp. 1–5, doi: 10.1109/IPACT.2017.8245126.

[11] M. Singh, Rajan M. A., Shivraj V. L., and Balamuralidhar P., "Secure MQTT for Internet of Things (IoT)," *Présenté à 2015 Fifth International Conference on Communication Systems and Network Technologies*, 2015, doi: 10.1109/CSNT.2015.16.

[12] A. Mektoubi, H. L. Hassani, H. Belhadaoui, M. Rifi, and A. Zakari, "New approach for securing communication over MQTT protocol A comparaison between RSA and Elliptic Curve," in *2016 Third International Conference on Systems of Collaboration (SysCo)*, 2016, pp. 1–6, doi: 10.1109/SYSCO.2016.7831326.

[13] Y. Upadhyay, A. Borole, and D. Dileepan, "MQTT based secured home automation system," in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, 2016, pp. 1–4, doi: 10.1109/CDAN.2016.7570945.

[14] S. Shin, K. Kobara, C. C. Chuang, and W. Huang, "A security framework for MQTT," in *2016 IEEE Conference on Communications and Network Security (CNS)*, 2016, pp. 432–436, doi: 10.1109/CNS.2016.7860532.

[15] M. Calabretta, R. Pecori, M. Vecchio, and L. Veltri, "MQTT-Auth: a Token-based Solution to Endow MQTT with Authentication and Authorization Capabilities," *Journal of Communications Software and Systems*, vol. 14, no. 4, pp. 320–331, Dec. 2018, doi: 10.24138/jcomss.v14i4.604.

[16] S. H. Ramos, M. T. Villalba, and R. Lacuesta, "MQTT Security: A Novel Fuzzing Approach," *Wirel. Commun. Mob. Comput*, vol. 2018, p. 1–11, 2018, doi: 10.1155/2018/8261746.

[17] A. Esfahani *et al.*, "A Lightweight Authentication Mechanism for M2M Communications in Industrial IoT Environment," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 288–296, Feb. 2019, doi: 10.1109/JIOT.2017.2737630.

[18] L. Malina, G. Srivastava, P. Dzurenda, J. Hajny, and R. Fujdiak, "A Secure Publish/Subscribe Protocol for Internet of Things," in *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES '19)*, Canterbury, CA, United Kingdom, 2019, pp. 1–10, doi: 10.1145/3339252.3340503.
[19] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," in *2017 IEEE International Systems Engineering Symposium (ISSE)*, Vienna, Austria, Oct. 2017, pp. 1–7, doi: 10.1109/SysEng.2017.8088251.
[20] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks," in *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, Bangalore, India, Jan. 2008, pp. 791–798, doi: 10.1109/COMSWA.2008.4554519.
[21] A. Banks and R. Gupta, "MQTT Version 3.1.1," Jt. Pap. Open Group OASIS OMG, 2014.
[22] A. Abderrahim, F. Ibtissam, C. Habiba, E. A. Hicham, and H. Nabil, "AES-Present: A New Secure IoT-Based Scheme for Telemedicine and e-Health Systems," *ARPN Journal of Engineering and Applied Sciences*, vol. 13, no. 24, pp. 9554–9559, 2018.
[23] A. Bogdanov *et al.*, "PRESENT: An ultra-lightweight block cipher," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2007, pp. 450–466, doi: 10.1007/978-3-540-74735-2_31.
[24] T. Yousuf, R. Mahmoud, F. Aloul, and I. Zualkernan, "Internet of Things (IoT) Security: Current Status, Challenges and Prospective Measures," *International Journal of Information Security Research (IJISR)*, vol. 5, no. 4, pp. 608 – 616, Dec. 2015.
[25] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Comput. Netw*, vol. 57, no. 10, pp. 2266–2279, Jul. 2013, doi: 10.1016/j.comnet.2012.12.018.
[26] S. Shin and K. Kobara, "Efficient augmented password-only authentication and key exchange for IKEv2," *IETF RFC 6628 Exp.*, 2012.
[27] I. Sahmi, T. Mazri, and N. Hmina, "Study of the Different Security Threats on the Internet of Things and their Applications," in *Proceedings of the 2nd International Conference on Networking, Information Systems & Security - NISS19*, Rabat, Morocco, 2019, pp. 1–6, doi: 10.1145/3320326.3320402.

BIOGRAPHIES OF AUTHORS

Imane Sahmi, Engineer degree in telecommunication systems from the National Institut of Posts and Telecommunications Rabat, Morocco, in 2010, she is an IT infrastructure at the National School of Applied Science kenitra since 2010, she's currently a Ph.D. student in the security of the Internet of Things since 2016 with the Systems Engineering laboratory.

Abderrahim Abdellaoui, Professor of Higher Education at National School of Applied Science - Kénitra, Ibn Tofail University (UIT), Morocco. Member of Engineering Sciences Laboratory. His current research interest includes Cloud Computing Security, Internet of things, and DNA cryptography.

Tomader Mazri, Professeur at the National School of Applied Sciences of Kenitra and a Permanent Member of the Electrical and Telecommunications Engineering Laboratory .HDR degree in Networks and Telecommunication from IbnTofail University, Ph.D. degree in Microelectronics and Telecommunication from Sidi Mohamed BenAbdellah University and INPT of Rabat, Master's degree in Microelectronics and Telecommunication Systems, Bachelor's degree in telecommunication from Cadi Ayyad University.

Nabil Hmina, Professor of Higher Education President of Sultan Moulay Slimane University since September 2018 to date, Director of the National School of Applied sciences 2011-2018, Degree in Physics, Option: Thermodynamics - Mohammed V University, University Ph.D. - Engineering Sciences, University, and Ecole Centrale de Nantes, 1994, HDR (1st in Morocco) Ibn Tofail University, Kenitra, 2002. Vice-President for Academic Affairs and Information Technology of the University Ibn Tofail, 2005-2011 Post-Doctoral Researcher: EDF - thermokinetics Laboratory of Nantes, 1994-1995, Research Engineer at PolyTech school of Nantes, 1995-1997. Director of the research laboratory "Systems Engineering" since 2011.