Journal on Internet of Things
DOI: 10.32604/jiot.2022.019893
Article

Tech Science Press

Lightweight Algorithm for MQTT Protocol to Enhance Power Consumption in Healthcare Environment

Anwar D. Alhejaili* and Omar H. Alhazmi

Department of Computer Science, Taibah University, Madinah, Saudi Arabia
*Corresponding Author: Anwar D. Alhejaili. Email: Anwaar_cs@hotmail.com
Received: 15 November 2021; Accepted: 15 February 2022

Abstract: Internet of things (IoT) is used in various fields such as smart cities, smart home, manufacturing industries, and healthcare. Its application in healthcare has many advantages and disadvantages. One of its most common protocols is Message Queue Telemetry Transport (MQTT). MQTT protocol works as a publisher/subscriber which is suitable for IoT devices with limited power. One of the drawbacks of MQTT is that it is easy to manipulate. The default security provided by MQTT during user authentication, through username and password, does not provide any type of data encryption, to ensure confidentiality or integrity. This paper focuses on the security of IoT healthcare over the MQTT protocol, through the implementation of lightweight generating and key exchange algorithms. The research contribution of this paper is twofold. The first one is to implement a lightweight generating and key exchange algorithm for MQTT protocol, with the key length of 64 bits through OMNET++ simulation. The second one is to obtain lower power consumption from some existing algorithms. Moreover, the power consumption through using the proposed algorithm is 0.78%, 1.16%, and 1.93% of power for 256 bits, 512 bits, and 1024 respectively. On the other hand, the power consumption without using the encryption is 0.25%, 0.51%, and 1.03% for the same three payloads length.

Keywords: Lightweight algorithm; IoT healthcare; MQTT; power consumption

1 Introduction

Significant advancement in the field of technology and communication has been witnessed around the world which leads to emerging IoT. IoT was proposed for the first time by Kevin Ashton in 1999; it has many different descriptions and definitions, but there is no single agreed upon definition. According to [1], the IEEE March 2014 report describes the IoT as the following: "A network of items—each embedded with sensors—which are connected to the Internet."

IoT provides numerous services and has a significant impact on many fields, such as smart cities, smart homes, industries, and healthcare. Currently, one of the most beneficial and popular IoT fields is

healthcare, especially in densely populated places, such as China [2]. According to [3,4], IoT healthcare can be defined as the infrastructure to facilitate the transmission and receipt of health data; it also allows better care via remote monitoring and improved decision making. The need for smart healthcare appeared for several reasons, including a tremendous increase in population, the emergence of many chronic diseases, lethal diseases that require continuous monitoring, and patients' need to be under continuous observation. In addition, the authors of [2] note that the future market for IoT healthcare services will grow rapidly; their forecast was based on statistics and analyses of IoT healthcare market from 2011 to 2016.

IoT healthcare enables real-time decisions based on the available data and collects more data using sensors. According to [5], smart healthcare aims to educate patients, keeping them aware of their health status continuously. This helps in decreasing the cost of continuous remote monitoring and saving time. One of the main benefits provided by the IoT in the health field is a rich user experience, with low cost and improved quality service. IoT healthcare allows doctors to expand their services regardless of geographic constraints.

In traditional hospitals, the local system is exposed to hackers or malfunctions, which may lead to loss of all or part of the patient data. However, smart healthcare can also be hacked, and medical devices can be hijacked. Although smart healthcare provides many contributions to increasing the efficiency of healthcare services provided around the world, there are some threats, as will mention.

IoT healthcare may be a double-edged sword that provides all services a patient may need, but it may also claim lives if wrong person tampers with it. Moreover, IoT leads to many types of security threats to the patient health. There are many risks that the IoT healthcare system may face, such as the security of communication, data integrity and availability. According to the authors of [6], the IoT medical devices, such as pacemakers, can be susceptible to a numbers of attacks, i.e., eavesdropping and spoofing. For pacemakers, an individual may send malicious commands, compromising the protection and causing direct physical harm to a patient [6]. Other research was focused on IoT devices for diabetes, and the authors of [7] mentioned some devices that transmit patient data without encryption, such as patient medication dosage.

There are also some restrictions regarding IoT devices which should be taken into consideration, such as limited computational power, battery life, and power issues. All the mentioned issues are considered serious problems and need solutions.

The IoT environment depends on the diversity of services and devices with limited resources. Due to these aspects, emergence of a lightweight protocol seemed appropriate for IoT devices. In addition, some of IoT protocols appeared as the following: Advanced Message Queuing Protocol (AMQP), Constrained Application Protocol (CoAP), Data Distribution Service (DDS), and Message Queue Telemetry Transport (MQTT). It is to be noted that all of the previous protocols are considered as application layer protocols, and the most common protocol is MQTT. MQTT protocol works as a publisher/subscriber which is suitable for IoT devices with limited power. One of the drawbacks of MQTT is that it is easy to manipulate. The default security provided by MQTT during user authentication, through username and password, does not provide any type of data encryption, to ensure confidentiality or integrity.

The aim of this paper is focuses on the security of IoT healthcare environment over the MQTT protocol, through the implementation of lightweight generating and key exchange algorithms.

The rest of this paper is organized as follows: Section 2 is an overview of MQTT protocol. Section 3 talk about some important studies for MQTT lightweight algorithms. Section 4 discusses the proposed

algorithm. Section 5 talk about how Implemented algorithm through OMNET++ simulation. Section 6 is the obtained Result. Section 7 talk about the discussion and limitation of the proposed algorithm, finally a conclusion and Future Work.

2 Message Queue Telemetry Transport (MQTT)

Regarding using many devices that must exchange data in real-time based on the internet connection and unreliable networks. Also, in case it should work with various protocols in data-link layer protocols, so it's appropriate to use MQTT protocol because it can work with various types of protocols such as Wi-Fi and Ethernet [8]. In 1999, Andy Stanford-Clark and Arlen Nipper developed the first version of MQTT, which was part of IBM's MQ Series message queuing product line. MQTT protocol considered as Publisher/Subscribe, work based on Broker that runs on top of (TCP/IP). According to the authors of [9] MQTT protocol can be worked over UDP through variant known as Message Query Telemetry Transport for Sensor Networks (MQTT-SN). Also, based on the study of [10] that showed for local and small connection better to use UDP, and for large network is better to use TCP. MQTT protocol is a lightweight messaging protocol between IoT devices, and a Machine-to-Machine (M2M) protocol. Moreover, it has the following characteristics: bidirectional connections (mean it can send data from device to cloud and back from cloud to device), Data agnostic (it does not care what type of data is sent), scalable, and designed for push communication. However, the default security in MQTT for authenticating between client and MQTT Broker through using username and password. Also, this authenticating process without any type of encryption.

MQTT work as Publish/Subscribe, where the publisher sends a message to all subscriber rather than a client-server architecture which the client needs to communicate with the end device directly. Also, in the MQTT protocol there is no direct communication between publishers and subscribers MQTT contains three main components are publisher, subscriber, and MQTT broker.

MQTT client (Publisher or Subscriber) creates an initial connection through creating a TCP/IP connection with the MQTT Broker. It uses the standard port for text in port number 1883. Also, it can use SSL/TLS encrypted communication port number 8883, but SSL/TLS could not be always a choice for all IoT devices. The following Fig. 1 illustrates the MQTT connection session.

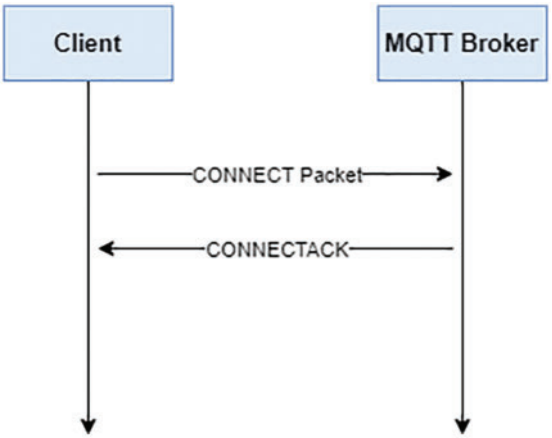

Figure 1: Client create a successful connection with MQTT

As illustrates in previous Figure the first contact with the Broker, the client needs to establish a connection with the Broker by CONNECT packet and get a response by CONNECT ACK. Also,

according to the authors of [11] the message that was sent or publish by byte format known as Payload. MQTT protocol support three different type of Quality of Service (QoS) to manage the MQTT content as the following [12,13]:

1. Quality of Service level 0 (QoS0), where the message is only sent once, and there is no delivery guarantee. Also, it known as "fire and forget".
2. Quality of Service level 1 (QoS1), where the message is only sent at least once, and it possible to deliver a message more than once. Also, it known as "acknowledged delivery".
3. Quality of Service level 2 (QoS2), where the message is sent once's exactly through using 4-way handshaking. Also, it known as "assured delivery".

However, each publisher need to publish in specific topic, and each subscriber will subscribe this topics. The topic which is the UTF-8 string containing different level as will illustrate in Fig. 2. The topics will be created by the subscriber while subscribe on it. Also, the subscriber can be subscribe more than one topic, but the publisher can only publish on one topic only.

hospital / second_floor / room02 / temperature

Figure 2: Example of MQTT topic architecture

3 Related Work

According to the authors of [14] in the digital world, privacy and trust allow effective data sharing. In sharing public healthcare data, transparency is a requirement for building confidence. Furthermore, one of the most significant challenges in the healthcare system is patient data privacy. Consequently, this leads us to think about the method of transferring data, algorithms, and protocols used in protection. One of the most popular and suitable protocols used in IoT healthcare is MQTT. MQTT protocol is used in an IoT communication device, but it has no security functions by default. To deal with the MQTT security problem, encryption/decryption and authentication such as username/password must be used, but it is useless and has high latency.

IoT device still needs lightweight algorithms to maintain the battery power device. This section will discuss some of the studies that proposed lightweight security algorithms for IoT devices based on an MQTT protocol.

The authors of [15] proposed a novel approach to achieve the confidentiality and integrity of data in an IoT device. This approach is known as the value-to-Keyed-Hash message authentication code (Value-to-HMAC). It is based on hash for the secret key. The secret key must be accepted by both the sender and the receiver. As a result, it has shown to be better than the symmetric key encryption algorithm. The authors of [16] proposed using digital signatures to distinguish abnormal traffic patterns of an IoT system. It is useful for detecting potential attacks on the IoT system. The authors of [16] apply this study on the MQTT protocols to the characterization of traffic. The authors of [17] proposed using digital signature with MQTT protocol but through Advanced Encryption Standard (AES) and Secure Hash algorithms (SHA). This study measured the overhead caused by using an encryption/decryption mechanism, through examining various types of AES.

The authors of [18] have compared the performance between DES, ECC, and TEA algorithms to determine the better lightweight encryption algorithm for IoT devices, based on MQTT protocol. As a result, the runtimes for the TEA, ECC, and DES algorithms were 1, 97, 119 ms, respectively. Consequently, the TEA encryption algorithm provided a satisfactory performance for IoT devices and can be considered as a lightweight encryption algorithm suitable for IoT devices.

In recently studies, the authors of [19] suggested key generation and distribution algorithm. It has a trusted system (TS), which is responsible for generating IDs and key for encryption. There is also a pre-shared secret (PSS) between publishers and subscribers. The results showed the difference between the nodes by using encrypted and unencrypted nodes; an increase in energy consumption by 2.71% was observed.

Deducing from the previous related work, most of the existing or proposed security algorithms contain complex calculations that lead to the device's power consumption. As a result, they are unsuitable for IoT applications. Therefore, this study will implement a lightweight key exchange algorithm to be suitable for IoT devices, and this will be our main focus in this paper.

4 The Proposed Algorithm

The following sections discuss the proposed generate/exchange secret key algorithm. Our algorithm is related to the ECDH concept but with a reduced overhead of computation. It uses prime number (PN), random number and hash function to ensure data integrity. Our proposed algorithms are based on the assumption of the TS. The TS can be a secure bootstrapping program that generates a PN for publishers and subscribers, after making sure they are authorized to enter according to their ID. However, to ensure security, the PN will be generated for each new session. It is same for the publisher and the subscriber for every session.

4.1 Generate and Exchange Stage

In this stage, after it is ensured that the publisher and subscriber are authorized to enter, the TS will generate a PN.

1. Send the PN, generated by TS, to the MQTT broker to be distributed to the publisher and subscriber.
2. The publisher and subscriber will have their own secret value called My_Key. The My_Key is generated by using random numbers. From 100 to 1, 000, it will be 3 digits.
3. Calculate the shared key for the publisher $SHp = (My\_Key \times PN)$ and the key for the subscriber $SHs = (My\_Key \times PN)$.
4. Then, after exchanging the key, the Secret_key will be calculated, which will be same value for both. The Secret_key for the publisher will be $S = (SHs \times My\_Key)$ and that for subscriber $S = (SHp \times My\_Key)$.

All these symbols are summarized in Tab. 1, and Fig. 3 shows the general flow of these steps.

Table 1: Symbols used in the proposed algorithm and their meanings

Symbol	Description
PN	Prime number for the publisher and subscriber.
My_Key	A separate unique random number for both the publisher and subscriber.
SHp	The publisher shared key, which equals ($PKp \times PN$).
SHs	The subscriber shared key, which equals ($PKs \times PN$).
S	The secret key for the publisher and subscriber.

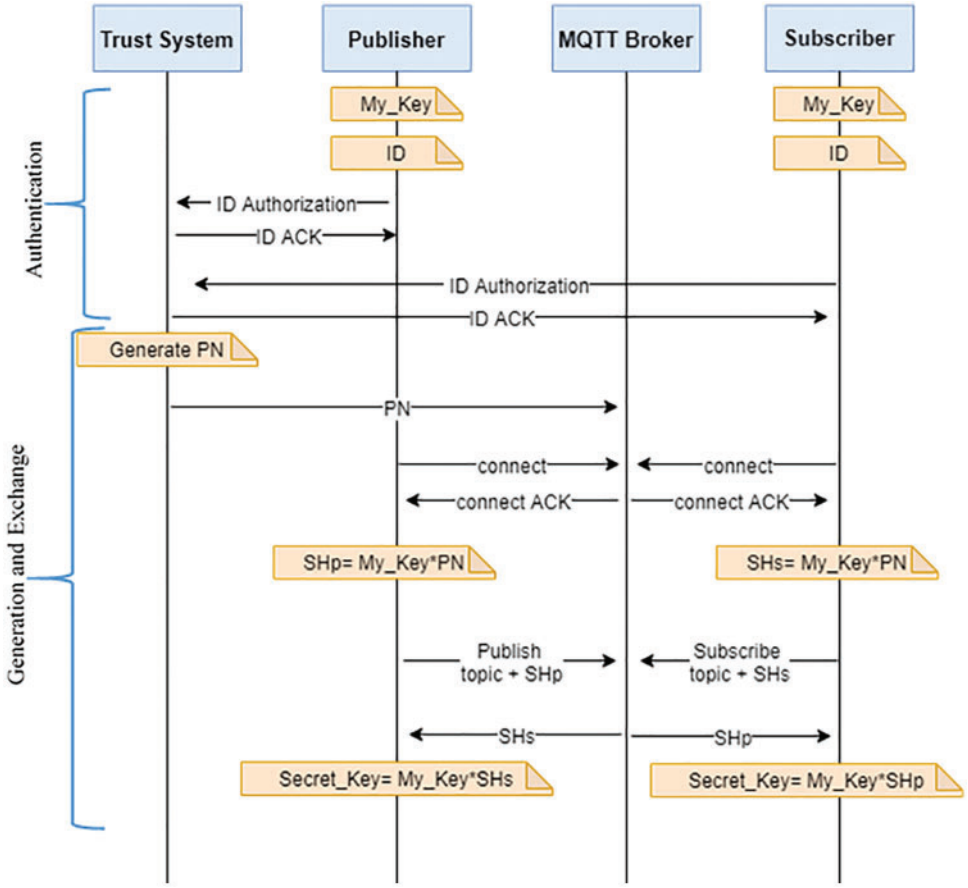

Figure 3: Flow of event of lightweight generating and key exchange algorithm

4.2 Encryption MQTT Payload Scenario Stage

Now, after generating and exchanging the secret key. The publisher and subscriber connect with the broker through CONNECT and get CONNECTACK as a response; the subscriber then subscribes to a specific topic. The secret key will be used to encrypt the MQTT payload. This will result in the following scenario:

The simple logical operation XOR will be used to reduce the overhead and power consumption. Also, it will use the Hash function to calculate the hash value of the payload before encrypting for payload integrity. Fig. 4. Illustrates the XOR encryption scenario.

5 Implementation

5.1 Test Environment

According to the authors of [20], there are two common types of simulation techniques: continuous simulation and Discrete Event Simulation (DES). This paper will use DES to implement the proposed algorithm over MQTT protocol. It is concerned with simulating the events of the systems during a specific period of time, as an example on it is OMNET++. This paper will incorporate OMNET++ for implementation.

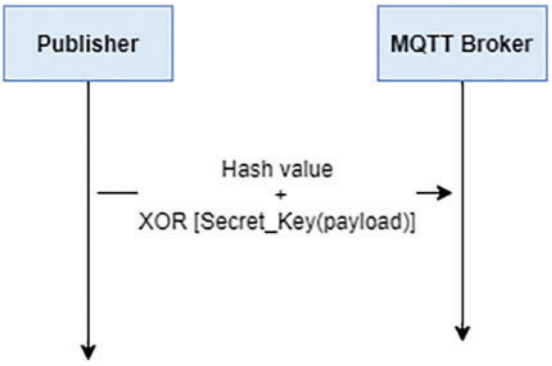

Figure 4: Example of MQTT topic architecture

The main elements in the network implemented on OMNET++ are four subscribers, four publishers, and one MQTT broker. All of these elements are considered essential as per the MQTT protocol. Cloud was available to store information. An additional element required for this algorithm is TS. It is responsible for verifying the IDs of both publishers and subscribers and then generate the PN. Furthermore, there are other network configuring elements, such as a router, access point, configurator, and radioMedium. We use a router facility to communicate between the TS server, MQTT broker server, and cloud server, which was linked by wire and access point (receiver). The use of configurator and radioMedium is necessary in OMNET++, for network configuration. Fig. 5 illustrates the network, where we use one access point, one router, and short distances between devices (estimated at 30 m), and MQTT protocol components. In this simulation, some essential points that were proposed; they are illustrated in Tab. 2.

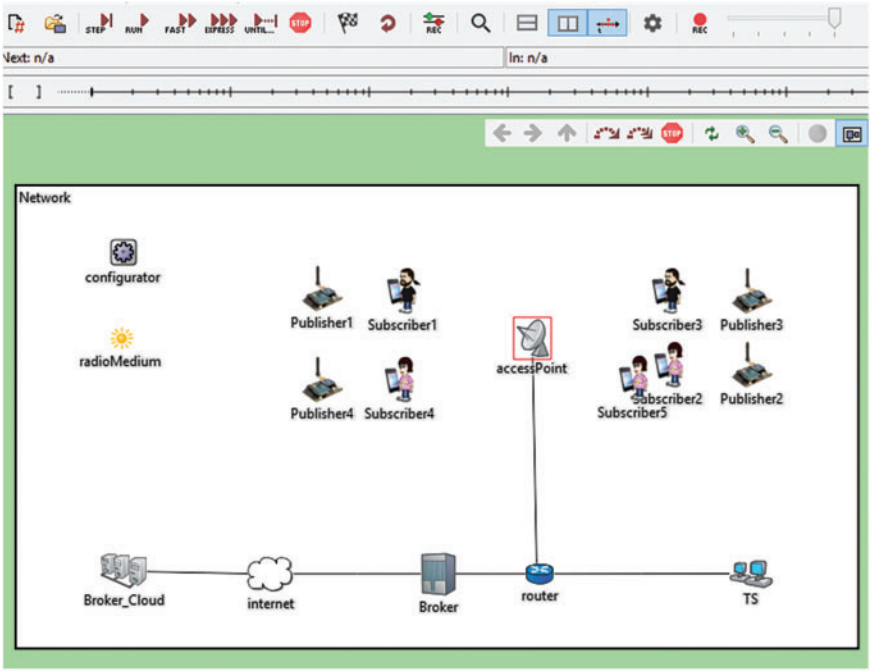

Figure 5: The network architecture

Table 2: Value of simulation parameter

Parameter	Value
Simulation time (run time)	2000 s
Distances	Less than 30 m
Delay	10 ms
Data rate	5 Mbps
Send interval	12 s

5.2 Implementation of the Algorithm

5.2.1 Stage1: Authentication

There is a unique ID for each device, which consists of 10 numbers and letters. It is to be noted that there is no relation between the ID of the device and the name of the device. The IDs of each devices will be generated based on random functions. The publisher or subscriber sends the ID to the TS, to verify, after which the TS will generate the PN.

5.2.2 Stage2: Generate and Key Exchange

This stage started with the TS generating a PN and sending it to the MQTT broker. Each of the publishers and subscribers have My_Key which be unique, secret and random generated. Also, each of the publishers and subscribers sends a connect massage and device information such as ID and device name to MQTT broker. The MQTT broker's task is to distribute the PN among publishers and subscribers, which is linked by ID devices. Then the shared key will be computed for both publisher and subscriber as discuss in Section 4.1. The MQTT broker will make the exchange shared key between publishers and subscribers, based on topics and ID. At the end, each publisher and subscriber requests the same topic as well as linked with a device ID that was stored in the broker has the same secret key. As shown in Fig. 6. if there is a topic that and has more than one subscriber, a secret key will be create for each of the subscribers and the publisher separately. There is an example in Fig. 6 where Publisher1 has two subscribers (Subscriber1 and Subscriber5) and two separate secret key were created.

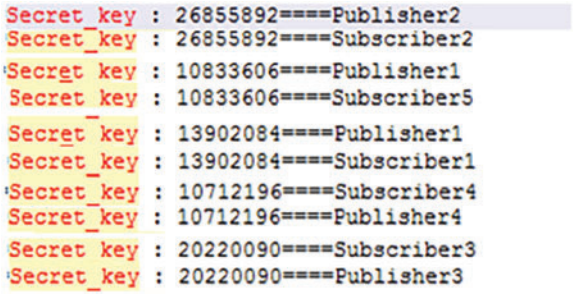

Figure 6: Secret key for publisher and subscriber based on topic and ID

5.2.3 Stage3: Encryption/Decryption

The encryption stage will occur while publish the topic. It will use the secret key with XOR operation, as shown in the code below, in Fig. 7, where the ecryptDecrypt contains the massage to

be encrypted. The decryption will occur at the subscriber's end, using the same secret key with XOR operation. Also, the Hash function will be used for the message before encryption, for data integrity.

```
std::string Publisher_App::encryptDecrypt(std::string toEncrypt) {
    std::string output = toEncrypt;
    for (int i = 0; i < toEncrypt.size(); i++)
        output[i] = toEncrypt[i] ^ Secret_key;
    return output;
}
```

Figure 7: The encryption/decryption code

6 Results

This section will mention and discuss the results obtained. The following results are based on simulation experience, during the simulation time of 2, 000 s. Also, simulation experience was done on three different packet lengths (256 bits, 512 bits, and 1024 bits). Also, the communication took place in a network between publishers and subscribers, using the ZigBee protocol. In this simulation, the power consumption was measured by using the proposed algorithm through the proposed architecture.

According to the authors of [21], the energy in the sensor can be measured through the receive and transfer packet. Measure the power consumption during receiving and transferring will require the use of the formula applied by the authors of [21,22].

The formula equation for power consumption while receiving the packet:

$$\boldsymbol{PS\_bits} \times \mathbf{50.0} \times \boldsymbol{NANO} \tag{1}$$

The formula equation for power consumption while transferring the packet:

$$\boldsymbol{PS\_bits} \times \mathbf{10} \times \boldsymbol{PICO} \times \boldsymbol{D} + \mathbf{50.0} \times \boldsymbol{NANO} \tag{2}$$

At the publisher's end, to measure power consumption during scanning will use the following formula equation:

$$5 \times \boldsymbol{NANO} \times \boldsymbol{Signals} \tag{3}$$

where the NANO value is 0.001 and PICO value is 0.000001. The PS_bits is referring to packet length by bits, which is assumed to be one of three different packet lengths (256 bits, 512 bits, and 1024 bits) each time. In formula Eq. (2), the D refers to distance, and in this network, it will be the range of ZigBee protocol, which equals to 30 m. In formula Eq. (3), signals value for each publisher is assumed to be equal to 1.

In the network, there are nine elements: five subscribers and four publishers; four different topics will be measured. Fig. 8 illustrated the average of power consumption for the previous nine elements by using the proposed encryption algorithm, and without the encryption.

The Y-axis in Fig. 8 represents the power of IoT devices by Joule, and the X-axis represents the packet length which are 256 bits, 512 bits, and 1024 bits over 2000 s. Fig. 8 shows the difference in power consumption between the proposed algorithm, which is generated, and the key exchange algorithm, with and without encryption. The Figure shows that the power consumption by the device is 0.78% when using the proposed algorithm for encryption with packet length 256 bits, while the power consumption of the device without encryption is only 0.25%, with same packet length. Also,

the power consumption by the device is 1.16% when using the proposed algorithm for encryption with packet length 512 bits, while the power consumption of the device without encryption is only 0.51%, with same packet length. Moreover, the power consumption by the device is 1.93% when using the proposed algorithm for encryption with packet length 1024 bits, while the power consumption of the device without encryption is only 1.03%, with same packet length. One can conclude that whenever the packet length is larger, the power consumption becomes larger too.

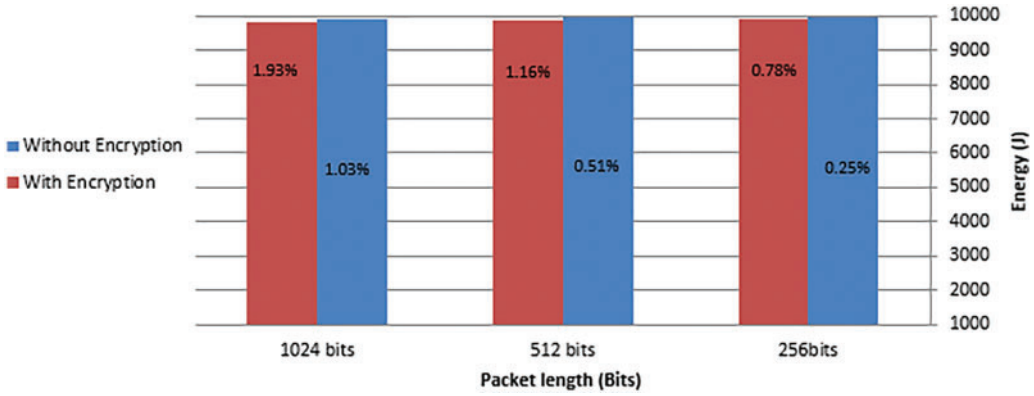

Figure 8: Power consumption by using encryption and without encryption

This shows the efficiency of the proposed algorithm for IoT devices through a lower amount of power consumption. Moreover, the percentage of power consumption by using the proposed algorithm through different packets length is better than on some existing algorithms, such as the algorithm that was suggested in [19], where the authors do not mention of the packet lengthas illustrated in Tab. 3. The authors of [19], proposed asymmetric cryptographic algorithm, using a different key to encrypt and decrypt. The subscribers, to decrypt the MQTT payload, will need to request cryptographic key from the TS. This was one of the reasons that led to an increase in the number of steps and power consumption.

Table 3: Comparison of power consumption with another algorithm

Proposed algorithm		Existing algorithm [19]	
Length	Percentage	Length	Percentage
256 bits	0.78%	Not specified	2.71%
512 bits	1.16%		
1024 bits	1.93%		

6.1 Security Analysis

This section will discuss the attacks that may affect the proposed algorithms.

- **Brute-force attack:** the attacker exploits the length of a key and tries to find all possible values to determine the key. The secret key size in proposed algorithm is equal to 64 bit. The number of possible values to get the secret key is 264 keys which equal 1.84×1019.

In addition, the average of possible number of tries to get the correct key will be greater than or equal to the half-possible number, which is $1.84 \times 1019/2$.

7 Discussion and Limitation

The result shown in Fig. 8 demonstrates that the performance of the proposed algorithm for lightweight generating and key exchange is suitable for IoT devices with limited power. From the performance point of view, the proposed algorithm leads to less power consumption than some of the existing algorithms even with long packet such as 1024 bits. Hence, the proposed algorithm is more reliable, as it provides data security through encryption and integrates by using a hash function. Additionally, TS is responsible to verify MQTT clients' IDs; this leads to providing authentication. The length of the key can be considered appropriate for such devices with limited power. The proposed algorithm can be extended and developed in the future after a few modifications.

There are some limitations of the proposed algorithm, such as the length of a key, which is 64 bits. On the other hand, an increase in the payload length leads to more power consumption, which may not be suitable for IoT devices.

8 Conclusion and Future Work

To improve security on the MQTT protocol and reduce the power consumption of the IoT device, the study proposed and implemented a lightweight algorithm for generating and exchanging keys, with a key length of 64 bits. The proposed algorithm comprises three major stages: authentication, lightweight generating and key exchange, and encryption/decryption. The authentication stage is based on the TS to verify MQTT clients' ID and then generate the PN. The second stage, which is lightweight generating and key exchange, involves computing a shared key and then the secret key that will be used in encryption. The third stage is based on encryption/decryption, using the secret key with XOR operation and hash function. All the algorithm stages have been implemented in an OMNET++ simulation during the simulation time of 2000 s. The simulation experience was done on three different packet lengths (256 bits, 512 bits, and 1024 bits).

In future work, XOR can be replaced by another lightweight and secure encryption algorithm, such as HIGHT block cipher [23] or PRESENT block cipher [24]. A different type of scenario attack can also be implemented, such as Man-in-the-middle attack (MITM) or SlowIT. SlowIT is a novel denial of service attack (DoS), and aimed at the MQTT protocol [25].

Funding Statement: Authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. H. Miraz, M. Ali, P. S. Excell and R. Picking, "A review on internet of things (IoT), internet of everything (IoE) and internet of nano things (IoNT)," *IEEE Xplore*, pp. 219–224, 2017.

[2] Y. Wan, J. He, H. Zhao, Y. H. Han and X. J. Huang, "Intelligent community medical service based on internet of things," *J. Interdiscip. Math*, vol. 21, pp. 1121–1126, 2018.

[3] I. Azimi, T. Pahikkala, A. M. Rahmani, H. Niela-Vilén, A. Axelin *et al.,* "Missing data resilient decision-making for healthcare IoT through personalization: A case study on maternal health," *Future Generation Computer Systems*, Elsevier, vol. 96, pp. 297–308, 2019.

[4] P. Sony and N. Sureshkumar, "Concept-based electronic health record retrieval system in healthcare IoT. Cognitive Informatics and Soft Computing," Springer, pp. 175–188, 2019.
[5] P. Sundaravadivel, E. Kougianos, S. P. Mohanty and M. K. Ganapathiraju, "Everything You wanted to know about smart health care: Evaluating the different technologies and components of the internet of things for better health. IEEE consumer electronics magazine", *IEEE*, vol. 7, no. 1, 2018.
[6] T. Flynn, G. Grispos, W. Glisson and W. Mahoney, "Knock! knock! who is there? investigating data leakage from a medical internet of things hijacking attack," in *Proc. of the 53rd Hawaii Int. Conf. on System Sciences*, 2020.
[7] S. Li, M. Zhang, A. Raghunathan and N. K. Jha, "Attacking and defending a diabetes therapy system. security and privacy for implantable medical device," Hawaii, Springer, vol. 9781461416746, pp. 175–193, 2014.
[8] H. Mrabet, S. Belguith, A. Alhomoud and A. Jemai, "A survey of IoT security based on a layered architecture of sensing and data analysis," *Sensors Switzerland*, vol. 20, no. 13, pp. 3625, 2020.
[9] M. H. Amaran, M. S. Rohmad, L. H. Adnan, N. N. Mohamed and H. Hashim, "Lightweight security for MQTT-SN," *International Journal of Engineering and Technology*, vol. 7, pp. 223–226, 2018.
[10] M. H. Amaran, N. A. M. Noh, M. S. Rohmad and H. Hashim, "A comparison of lightweight communication protocols in robotic applications," *Elsevier Procedia Computer Science*, vol. 76, no. Iris, pp. 400–405, 2015.
[11] M. Bender, E. Kirdan and M. Pahl, "Open-source MQTT evaluation," *Proceedings of IEEE 18th Annual Consumer Communications and Networking Conference*, Las Vegas, USA, pp. 1–4, 2021.
[12] D. S. Ugalde, "Security analysis for MQTT in internet of things security analysis for MQTT in internet of things," M.S. thesis, Kth Royal Institute of Technology School of Electrical Engineering and Computer Science, SWEDEN 2018.
[13] D. Soni and A. Makwana, "A survey on mqtt: A protocol of internet of things (IoT)," in *Int. Conf. on Telecommunication, Power Analysis and Computing Techniques (Ictpact-2017)*, Chennai, India, no. April, 2017.
[14] L. X. Fadrique, "Privacy and trust in healthcare IoT data sharing: A snapshot of the users' perspectives," M.S. thesis, University of Waterloo, Waterloo, 2019.
[15] D. Dinculeană and X. Cheng, "Vulnerabilities and limitations of MQTT protocol used between IoT devices," *Applied Sciences (Switzerland)*, vol. 9, no. 5, pp. 848, 2019.
[16] R. Leal, L. Santos, R. Vieira, R. Gonaalves and C. Rabadao, "MQTT flow signatures for the internet of things," in *Iberian Conf. on Information Systems and Technologies*, Yalta, Russia, CISTI, Vol. 2019-June, 2019.
[17] A. Fauzan, P. Sukarno and A. Wardana, "A. overhead analysis of the Use of digital signature in MQTT protocol for constrained device in the internet of things system," in *2020 3rd Int. Conf. on Computer and Informatics Engineering*, Changsha, China, IC2IE, 2020.
[18] Y. Pan, X. Cheng and C. Xia, "Research and design of lightweight encryption for Mqtt protocol," *Mechanical and Control Engineering (MCE) Contents List Available at VOLKSON PRESS*, vol. 1, no. Icid, pp. 143–145, 2017.
[19] A., Bashir and A. H. Mir, "Lightweight secure-MQTT for internet of things," *Optical and Wireless Technologies*, pp. 57–66, 2020.
[20] R. García and L. E, "Performance evaluation by simulation and analysis with applications to computer networks," *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 1–12, 2013.
[21] Y. Zhou and G. Li, "Convergence of communication and computing in future mobile communication systems," *Telecommunications Science*, vol. 34, no. 3, pp. 1–7, 2018.
[22] S. Ramadhani, "Energy efficient wireless sensor network for monitoring temperature and relative humidity in forest," Ph.D. dissertation, Nelson Mandela African Institution of Science and Technology, Arusha, 2020.

[23] D. Hong, J. Sung, S. Hong *et al.,* "HIGHT: A new Block Cipher Suitable for low-Resource Device," *Int. Workshop on Cryptographic Hardware and Embedded Systems*, Springer, Berlin, Heidelberg, pp. 46–59, 2006.

[24] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar and Poschmann, "PRESENT: An ultra-lightweight block cipher," *International Workshop on Cryptographic Hardware and Embedded Systems*, Berlin, Heidelberg, Springer, pp. 450–466, 2007.

[25] I., Vaccari, M. Aiello and E. Cambiaso, "SlowITe, a novel denial of service attack affecting MQTT," *Sensors (Switzerland)*, vol. 20, no. 10, pp. 2932, 2020.