EAI Endorsed Transactions
on Internet of Things

Research Article EAI.EU

Securing Communication in MQTT enabled Internet of Things with Lightweight security protocol

Adil Bashir and Ajaz Hussain Mir

Department of Electronics and Communication Engineering
National Institute of Technology Srinagar, Jammu and Kashmir, India-190006
Email: adilbashir.445@gmail.com

Abstract

This paper proposes a security algorithm for Internet of Things (IoT) using simple lightweight cryptographic operations. The main advantage of the proposed algorithm is the simplicity, energy efficiency and the speed of algorithm such that it can be computed quickly using a low-power microcontroller. The encryption of the sensed data is performed using simple operations so as to consume smaller amount of node energy. To test the effectiveness, of the proposed algorithm, an experimental rig is set up to implement the proposed algorithm. The analysis confirms that the proposed algorithm provides end-to-end encryption and imparts security against likely attacks such as brute force attack, spoofing attack, and has small code footprint. It is envisaged that the algorithm can be very useful in securing message transmissions in Internet of Things.

Keywords: Internet of Things (IoT), Constrained Application Protocol (CoAP), Message Queuing Telemetry Transport (MQTT), IoT Security, MQTT Broker.

Received on 17 June 2017, accepted on 08 August 2017, published on 06 April 2018

doi: 10.4108/eai.6-4-2018.154390

1. Introduction

Internet of Things (IoT) is rapidly gaining popularity due to its potential to bring global digital revolution and is being increasingly used in industrial, transportation, digital health and agricultural applications [1]. With numerous applications, IoT brings lot of research challenges that include having a requisite architecture for control and communication among devices [2] Miniaturization of Components [3], Interoperability [4], Service Orchestration [3], Security & Privacy [5, 6], Standardization [7], etc. IoT being resource constrained network demands lightweight protocols at each layer of Internet Engineering Task Force (IETF) defined protocol stack [8]. The application layer protocols are considered as building-blocks to achieve the expected requirements (e.g. scalability, reliability, performance) in Internet of Things environment. The commonly used protocols at application layer of IoT are Constrained Application Protocol (CoAP) and Message Queuing Telemetry Transport (MQTT) being simple and lightweight protocols.

Among the research challenges presented above, security and privacy is considered as one of the main hindrance for the widespread growth and adoption of IoT as there is no public confidence that IoT will not cause harm to user privacy. The security algorithm for IoT need to take care of several issues that include protecting user privacy, consuming less energy for the process and providing strong security against attacks.

Security mechanisms at each layer of IoT protocol stack [8, 2] are implemented to safeguard the sensed information from adversaries. For example, at link layer, encryption algorithms at the hardware in IEEE 802.15.4 sensing platforms are used to provide security features, for instance in TelosB motes, Advanced Encryption Standard (AES) is used as a symmetric cryptosystem at link layer. However, the encryption mechanisms at link-layer only safeguards hop-to-hop communication and the messages on an IP network are not protected from adversaries. At network layer, IPSec [9] is used to provide security related services. But employing IPSec induces

packet overhead that requires more processing power and time. Further the protocols used in IPSec are complex structured and consume lot of device energy. The application layer protocols rely on the security protocols at the underlying transport layer for providing security elements to the communicating messages. One such protocol at the transport layer is Transport Layer Security (TLS) that guarantees security between applications and provides the features like host authentication, data confidentiality and integrity. However, TLS is only used over Transmission Control Protocol (TCP) that is rarely used in IoT. A variant of TLS, for widely used protocol User Datagram Protocol (UDP), known as Datagram Transport Layer Security (DTLS) is available, but it is not commonly used for IoT because of high data loss rates due to packet fragmentation [10].

From the overview, it can be deduced that security protocols discussed above are actually the modified versions of existing complex network security protocols and the modified versions are still complex for IoT. These protocols are difficult to implement on resource constricted IoT nodes as involved cryptographic operations can be expensive in terms of code size and processing speed.

The aim of this paper is to enable application layer protocol MQTT with security elements so as to encrypt payload of communicating messages with the proposed lightweight secure cryptographic algorithm. We consider MQTT as application layer protocol because it is itself lightweight and popular protocol for IoT and very less work has been done to secure MQTT messages at application layer. The key benefits of the proposed security algorithm are energy efficiency, simplicity and secure. The idea is based on using less computational structure of the cipher so that the available limited energy at nodes will last for long to keep device operational for longer time (to make it energy efficient). Unlike other approaches thus far which rely on conventional network security protocols, our method is designed in view of resource constrained nature of IoT devices

2. Related Work

A lattice-based cryptosystem "NtruEncrypt" is known to be alternative of RSA cryptosystem. [11] performed the comparative analysis of NtruEncrypt, ECC and Rabin's scheme that were candidate security protocols to be used for resource constrained environments like Wireless Sensor Networks and Internet of Things. The results of comparative analysis depict that NtruEncrypt consumes less average power in comparison to other two protocols. However, it requires large sized messages that might lead to packet fragmentation and increase the packet re-transmission rate due to communication errors.

Authors in [12] used Attribute-Based Encryption (ABE) scheme to ensure security for IoT based on Publish-Subscribe (Pub-Sub) architecture. To encrypt messages, symmetric Advance Encryption Standard (AES) cryptography is used and the AES key itself is encrypted using ABE scheme. Since most of the IoT devices have limited resources and generate smaller number of data bits, therefore using highly computational AES and ABE cryptographic techniques for smaller number of generated bits is not suitable. [13] employed Predicate Based Encryption and CP-ABE to protect published content and the privacy of the client's (subscriber's) interested topics.

IPSec is used at the network layer with Internet Key Exchange (IKE) Protocol [14] to impart security associations (SAs) among nodes. IKE provide mutual authentication by implementing RSA based-certificates. Authors in [14] propose Elliptic Curve Cryptography (ECC) based IKE for mutual authentication and Elliptic Curve-Diffie Hellman (ECDH) based key agreement scheme in lieu of RSA and DH algorithms. The scheme lowers the computation cost and needs smaller sized key for the similar level [15]. IKE scheme is based on certificates and thus unsuitable for miniature IoT devices. A different security mechanism known as Identity Based cryptographic Scheme (IBS) does not rely on certificates from third party, thus making it advantageous for IoT. IBS was first developed and implemented by Shamir [16]. In resource constrained devices, Identity-Based Encryption (IBE) is designed and implemented using RSA, Elgamal protocols [17], however, these are found to be highly computational for resource constricted nodes due to their exponentiation operations and hence making them inappropriate for IoT.

[18] presented DTLS as a security protocol for IoT that is placed between application layer and transport layer. DTLS is based on RSA and is developed for Low Power Networks. The analysis of this scheme depicts that DTLS imparts confidentiality, integrity and authentication, but consumes significant device energy and induce overheads.

3. Proposed Method

This section describes the proposed algorithm to safeguard MQTT messages from malicious users. The aim of the proposed algorithm is to prevent sensed data from illegitimate users by employing a lightweight encryption algorithm. Being lightweight, it utilizes less node energy for computations, therefore increasing the longevity of IoT nodes. This is, as far as our knowledge goes, the first security algorithm for encrypting MQTT payload.

3.1 Application Scenario

The application scenario that has been considered is IoT based Home Automation System consisting of a resource-rich IoT device that acts as server (S), three IoT nodes (I1, I2, I3) acting as sensor devices (MQTT clients i.e. Publisher & Subscribers) and MQTT broker (see Figure 1). In our experimental setup, one Raspberry Pi

[19] device is used as a server and other Raspberry Pi is used as MQTT broker. MQTT clients are equipped with sensors such as temperature sensor, humidity sensor, Passive Infra-Red (PIR) sensor and have ESP8266 [20] as communication equipment. MQTT clients have constrained resources in terms of energy, memory and computational power, therefore the available limited node resources need to be utilized efficiently. Contrasting MQTT clients, server and broker devices have no such limitations in terms of processing power, energy and memory capacities as Raspberry Pi has 900MHz quad-core ARM Cortex-A7 processor, 1GB RAM, supports 128GB SD card and is connected to uninterrupted power supply in our scenario. Server is responsible for assigning unique identifiers (IDs) to every IoT client node. For each session, server assigns different IDs to client nodes, therefore, making it difficult for adversary to guess the value of node ID at particular instant of time as the assigned IDs are short lived. All nodes contain pre-shared secret (PSS) that is stored securely during the registration process using a secure bootstrapping as specified in [21].

There are three phases of algorithm execution i.e. Key generation & distribution phase, Encryption & Publish phase and Decryption phase. In key generation and distribution phase, server generates the key and shares with communicating IoT clients. During Encryption and publish phase, the sensed data is encrypted by derived key at IoT client and then the encrypted data is published on the topic and sent to MQTT broker. In decryption phase, subscriber receives the data on given subscribed topic from MQTT broker and then uses decryption algorithm with derived key from S. The proposed system architecture is shown in the figure 1.

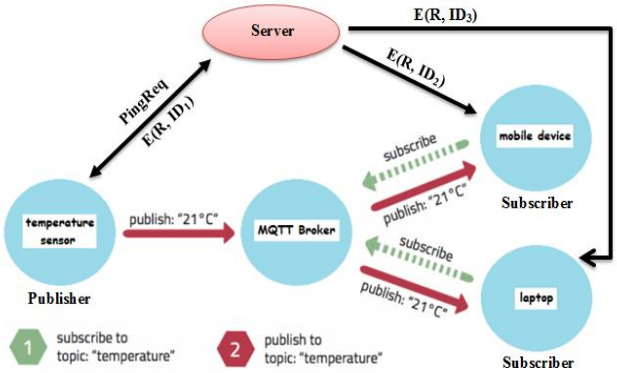

Figure 1. IoT based Home Automation System

In figure 1, I1 (Temperature Sensor) is publisher device and I2 (mobile device) & I3 (laptop) are subscriber devices. Whenever publisher (I1) device senses an event (temperature), it sends PingReq message to server which then sends the encryption key to it. This key is used to encrypt the temperature data at I1 which is then forwarded to MQTT broker. The subscribers receives decryption key from server and deciphers received data from MQTT broker.

3.2 Proposed Algorithm

This section describes the proposed algorithm that is implemented to safeguard the sensed data at IoT client nodes. As described above in the application scenario, I1 acts as publishing node and has sensed an event to be transmitted to subscribers i.e. I2 & I3. The following algorithmic steps will occur as soon as I1 has data to be published.

I1 will send "PingReq" message to S. S will run Algorithm A to generate secret key for secure communication between I1 and I3, I2. Here, we will consider I3 only as subscriber device to simplify the explanation. Our proposed algorithm consists of three phases as defined below:

Key Generation and Distribution Phase

This phase is performed at server that implements algorithm A for generating and sharing IDs and secret keys.

Algorithm A:

Step 1: Execute ID generation algorithm and distribute IDs to every IoT client node.

Step 2: Generate a set (s) of random numbers by Pseudo-Random Number Generator (PRNG) using seed S1 and group them into 'b' blocks.

Step 3: Generate 'b' number of random integers to be used as block selector.

Step 4: Generate 'n' number of random integers to be used as number selector from the chosen block (n = s/b)

Step 5: Select the block randomly from step 3 and select the number (R) arbitrarily from chosen block. This R will act as encryption key.

Step 6: S will encrypt key (R) using device ID of publisher (I1 in our scenario) and subscriber (I3 in our scenario) and will share the generated key with I1 and I3 respectively i.e.

K1 = E(R, ID1) --- for Publisher

K3 = E(R, ID3) ---for Subscriber

Encryption and Publish Phase

This phase runs at publisher node. The data sensed by publisher is encrypted by the derived key from server. The encrypted data is published on the topic and send to MQTT broker. The encryption is performed using algorithm B described below.

Algorithm B:

Step 1: I1 will decrypt K1 using its device ID (ID1) to get encryption key 'R'.

R = D(K1, ID1)

Step 2: I1 will encrypt sensed data (m) using key 'R'.

ED = m ^ R

Step 3: Publish ED to MQTT broker.

Decryption Phase

This phase runs at subscriber node, which after obtaining key from server and data from broker launches algorithm C to decipher the encrypted data. The algorithm at subscriber is:

Algorithm C:

Step 1: I3 will get K3 from the server 'S'. I3 will decipher it using its device ID (ID3) to get key 'R'.

$$R = D(K3, ID3)$$

Step 2: Decrypt ED using k.

$$m = ED \wedge R$$

Subscriber decrypts the message received from MQTT broker and performs the required action based on the message contents. The overall procedure of our proposed algorithm is represented in figure 2.

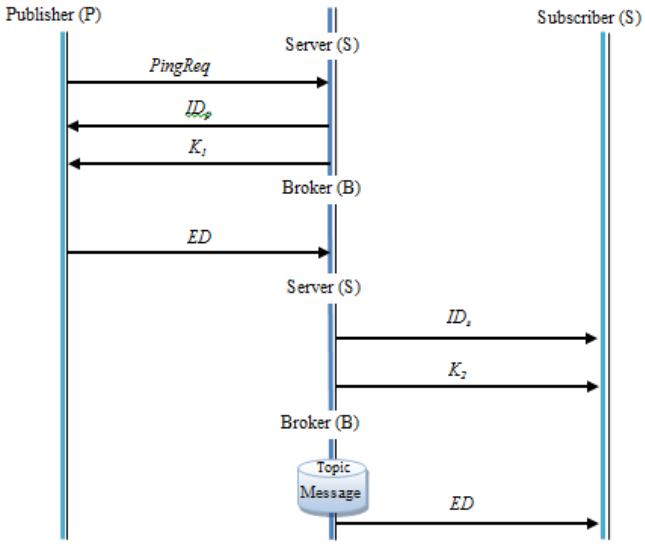

Figure 2. Message Event Sequence

In the initialization phase, Publisher (P) has sensed data that it wants to transmit to the subscriber via broker. Publisher will send Ping message (PingReq) to server that indicates that it has some data to be forwarded to subscriber. Server will respond to Ping message (PingResp) by generating device IDs for all IoT nodes for the current session. Further, the server runs Algorithm A to generate unique random number (R) to be used as encryption key. R is encrypted using publisher's device ID (IDp) generated for current session and send to Publisher (K1 = E(R, IDp)). Publisher decrypts K1 to get the key to be used for encrypting sensed data. The encrypted data (ED) is sent on the given topic to the MQTT broker. At the same time, server sends encrypted key (K3 = E(R, IDs)) to the subscriber. At this time, the broker forwards encrypted data (ED) to the subscriber which decrypts the K3 and then uses it to decipher the data received from broker. In this way, the MQTT message travels securely from publisher to the subscriber.

3.3 Implementation

The algorithm has been implemented in experimental set-up consisting of IoT client nodes, server and broker devices. IoT nodes are designed to have arduino nano with ATMEGA 328P [22] as processing unit, DHT11 [23] as temperature sensor, ESP8266 [20] as communication equipment. The view of the IoT node is shown in figure 3 below:

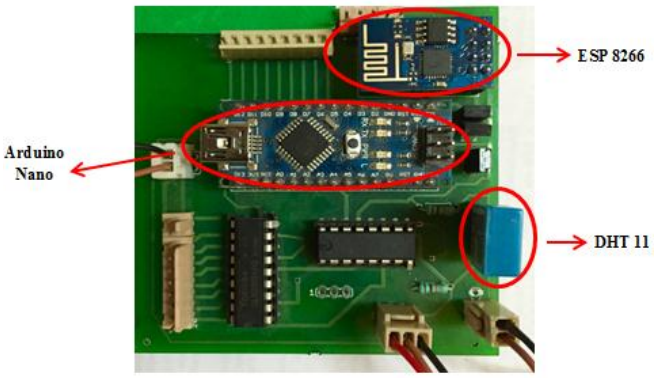

Figure 3. Exemplary IoT Node

Technical parameters of the components of exemplary IoT node are:

1. Arduino Nano - based on the ATmega328P microcontroller AVR architecture (Flash Memory of 32 KB, clock 16 MHz, 8 Analog I/O Pins, 22 Digital I/O Pins, 1 KB EEPROM)
2. ESP8266 (Wi-Fi communication device)
3. 9 volt battery (Power supplier)
4. DHT11 (Temperature sensor)

The overall view of our experimental setup is shown in figure 4.

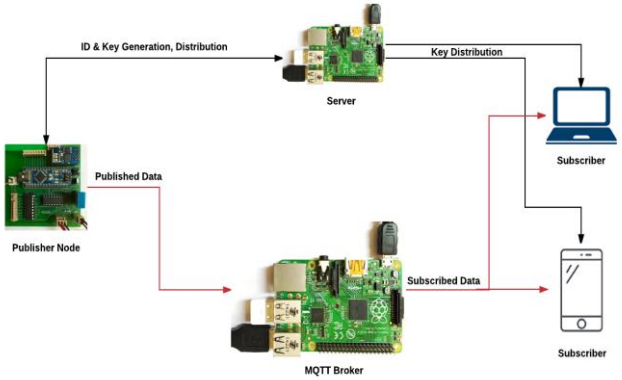

Figure 4. Experimental Set-up

4. Analysis of Proposed Algorithm

4.1 Security Analysis

End-to-End Encryption

The algorithm provides end-to-end encryption for communicating messages in IoT network. This is explained in a way that publisher encrypts message and sends encrypted message to the broker which forwards it to the subscriber, only which can decrypt the message. Broker is privileged to have messages from publishers but it cannot access the actual message contents as it doesn't have keys to decipher the messages, therefore, message confidentiality is guaranteed even in the presence of broker hijack attack. It is also deduced from the analysis that the proposed algorithm provides security against malicious broker because the deciphering keys are only

possessed by subscribers. So in this way our algorithm provides end-to-end encryption.

Brute force Attack

The algorithm is resilient to brute force attack. In our algorithm, the key size is 32 bits, therefore the number of possible values for key 'k' is

$$n = 232*(1/2) = 4.29 *109*(1/2)$$

The number of possible values for key also indicates the number of guesses that an attacker has to make to get the value of encryption key using brute force attack. Thus, the probability that an adversary conjectures the value of key is

$$p = 1/ n$$

i.e. $$p = 1 / (4.29 *109*(1/2))$$

The value of p indicates that the probability of conjecturing the value of key is very small. Therefore making the algorithm resilient to brute force attack which can further be made stronger by using large sized key.

Spoofing Attack

If the adversary pretends to be server (S), the adversary can get information about generated session IDs and cryptographic keys. This assists adversary to decipher all communicating messages among IoT nodes. Our proposed algorithm counters this attack by using pre-shared secret (PSS) that is set-up during bootstrapping phase. Publisher, that has data to be transmitted, encrypts "PingReq" message with PSS before forwarding it to server for obtaining encryption key. Illegitimate server can't decipher PingReq message as it does not have PSS and thus cannot access the encrypted published messages also.

4.2 Performance Analysis

Time Complexity

The time complexity of the proposed algorithm is calculated below:

$$T(n) = 6*O(1) + 2*O(11) + 8*O(10) + 4*O(1)$$

As T is small, therefore the time complexity becomes constant.

Storage Requirements

The implementation of the proposed method requires approximately 26 KB which is suitable code size for IoT nodes. For instance, in our application scenario, IoT nodes have ATmega328P microcontroller that provides 32KB ROM space which is sufficient for our algorithm. Similarly, IoT motes from different vendors have ROM space more than what is required by our algorithm, for example, [24] that have 128KB ROM space which is adequate to hold our algorithm. Due to small code footprint, there is a provision to increase key size to make algorithm more resilient against attacks.

5. Conclusion

Security and privacy is one of the major challenges for widespread adoption of Internet of Things. Numerous devices enabled with sensors will be deployed for gathering information from the environment which will be shared with other devices and servers. The information gathered by these miniature devices can lead to unprecedented privacy loss. Existing network security protocols like AES, RSA, etc. being computationally expensive can't be employed in IoT.

In this paper, a lightweight encryption algorithm is proposed to secure confidential messages exchanged among IoT nodes. It is shown that the proposed algorithm is simple, secure against popular attacks and is memory efficient. IoT demands efficient utilization of node resources that include energy, memory and CPU. To improve the lifetime of IoT nodes, less computational security algorithm is proposed that utilizes node energy efficiently. The proposed algorithm is a good choice for IoT and it overcomes attacks like brute force attack, spoofing attack and also provides end-to-end encryption. The memory footprint of the proposed algorithm makes it suitable for resource constrained IoT nodes. It is also deduced from the analysis of the proposed algorithm that it does not induce computational overhead and is energy efficient which makes it suitable for Internet of Things.

References

[1] R. KHAN ET AL. (2012) Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges. In IEEE International conference on Frontiers of Information Technology (FIT).

[2] J. GRANJAL ET AL. (2015) Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. In IEEE communication and surveys tutorials, vol. 17, no. 3.

[3] Internet of Things: Challenges and Issues. URL http://www.cse.wustl.edu/~jain/talks/iot_ad14.htm. [visited on 2017-07].

[4] I. ISHAQ ET AL. (2013). IETF standardization in the field of Internet of Things (IoT): A survey. In J. Sens. Actuator Netw., vol. 2, pp. 235-287.

[5] A. JULES (2006). RFID security and privacy: a research survey. In IEEE journal on selected areas in communications 24 (2) 381-394.

[6] J. BUKLEY (2006). From RFID to the internet of Things: final report. In European Commission Conference Brussels, Belgium.

[7] L. ATZORI ET AL (2010). The Internet of Things: A Survey. In Computer Networks, vol. 54, no. 15, pp. 2787-2805.

[8] M. PALATTELLA ET AL (2103). Standardized Protocol Stack for the Internet of (Important) Things. In Communications Surveys & Tutorials, IEEE, 15 (3) 1389-1406.

[9] S. RAZA ET AL (2011). Securing Communication in 6LoWPAN with Compressed IPsec. In IEEE Internation Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS).

[10] K. HARTKE (2014). Practical Issues with Datagram Transport Layer Security in Constrained Environments, IETF draft-hartke-dice-practical-issues-01.

[11] G. GAUBATZ ET AL (2005). State-of-the-art in ultra-low power public key cryptography for wireless sensor

networks. In 3rd IEEE International Conference on Pervasive Computing and Communications Workshop (PERCOMW).
[12] X. WANG ET AL (2014). Performance evaluation of Attribute-Based Encryption: Toward data privacy in the IoT. In IEEE International Conference on Communications (ICC) pp. 725–730.
[13] P. PAL ET AL (2012). P3S: A Privacy Preserving Publish-subscribe Middleware. In Proceedings of the 13th International Middleware Conference pp. 476–495.
[14] S. RAY ET AL (2012). Establishment of ECC-based initial secrecy usable for IKE implementation. In Proceedings of World Congress on Expert Systems (WCE).
[15] NSA, The Case for Elliptic Curve Cryptography. URL http://www.nsa.gov/business/programs/elliptic_curve.shtml (visited on 2014-02).
[16] A. SHAMIR (1984). Identity-based cryptosystems and signature schemes. In Proceedings Crypto'84, Santa Barbara, California, USA, pp. 47–54.
[17] C. GENTRY (2006). In Practical identity-based encryption without random oracles. In Proceedings of the EUROCRYPTO'06, Springer-Verlag, pp. 445–464.
[18] J. PARK ET AL (2014). Lightweight Secure Communication for CoAP-enabled Internet of Things using Delegated DTLS Handshake. In IEEE International conference on Information and Communication Technology Convergence (ICTC).
[19] Raspberry Pi 2 Model B. URL https://www.raspberrypi.org/products/raspberry-pi-2-model-b. [visited on 2017-07].
[20] ESP8266 WiFi Module. URL https://www.sparkfun.com/products/13678. [visited on 2017-07].
[21] N. KANG ET AL (2014). Secure initial-key reconfiguration for resource constrained devices, IETF draft-kang-core-secure-reconfiguration- 01.
[22] ATmega328P microcontroller. URL http://www.microchip.com/wwwproducts/en/ATmega328P [visited on 2017-07].
[23] Temperature Sensor. URL https://playground.arduino.cc/Main/DHT11Lib. [visited on 2017-07].
[24] Zolertia motes. URL https://zolertia.io/product/re-mote. [visted on 2017-07].