A Lightweight Multi-tier S-MQTT Framework to Secure Communication between low-end IoT Nodes

Abdur Rahman[*], Shanto Roy[†], M Shamim Kaiser [‡] and Md. Shahidul Islam [§]
Institute of IT, Jahangirnagar University[*†‡§], Dhaka-1342, BD
Dept. of CSE, Green University of Bangladesh[†], Dhaka-1207, BD
Email: abdurrahman.iit[*]@gmail.com, shantoroy[†]@ieee.org, mskaiser[‡]@juniv.edu, sislam[§]@juniv.edu

***Abstract*—The evolution and expansion of networking technologies have managed to create large scale connectivity among versatile devices and applications that led to the jargon internet of things (IoT). IoT has evolved due to the convergence of wireless sensor networks (WSN) and internet technologies with a view to approaching towards smart city prospects. In IoT, for maintaining device to device communication, HTTP protocol has been used for remote monitoring and analysis of data from large number of sensing elements but it consumes more power, have comparatively lesser efficiency of transmission and cannot utilize system bandwidth efficiently as well. Thus the protocols MQTT (Message Queuing Telemetry Transport), AMQP and CoAP are quite capable of handling wireless sensor traffic under very low bandwidth and constrained network conditions. Security is also another major concern as IoT applications collect private data and allow access to various control functions over the internet. Therefore, in this paper, we discuss a detailed analysis of data & devices security issues and present an enhanced security model with a view to improving the security issues. We propose a secure version of MQTT protocol modifying and enhancing the existing MQTT protocol based on Key/Cipher text Policy Attribute Based Encryption(KP/CP-ABE) using lightweight Elliptic Curve crypto-system. We also introduced a multi-tier authentication system for secure communication and an extra security layer to prevent the data theft.**

keywords- IoT Data Security, Secure MQTT protocol, Key/Cipher text Policy Attribute Based Encryption(KP/CP-ABE), Elliptic Curve Cryptography, Lightweight Secure communication

I. Introduction

IoT has led to a highly converged inter-networking for different sensor devices and networks that comes with various applications including manufacturing [1], smart grids [2], online healthcare [3], agriculture [4], and transportation systems [5] etc. that enriching the smart city aspects. As different devices and applications are connected altogether creating a complex wireless sensor network, IoT communication needs to be both lightweight and secure in order to achieve high efficiency and performance.

IoT is improving the way of automation and communication day by day at a tremendous rate. As a result to ensure optimization in energy, performance and quality, the system needs to be designed as much lightweight as it can be. Therefore, MQTT protocol is a lightweight data centric communication protocol that can be run on simple battery operated low energy consuming sensor nodes [6]. MQTT has been ensuring and maintaining a satisfactory energy efficiency in low end wireless sensor networks that makes this protocol suitable for industrial usage purpose in IoT arena.

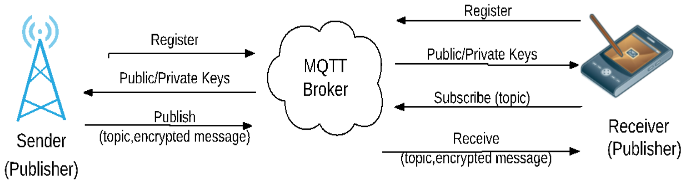


Fig. 1. System diagram

Having a lot of advantages using lightweight systems, still there are several security and privacy concerns at different layers that prevent a full adoption of IoT. This is because the major challenges in IoT is related to security of data and infrastructure. Therefore, our proposed secure version of MQTT protocol and multi-tier authentication system ensure security and privacy of sensitive data & devices. In order to achieve this particular goal, we used KP/CP-ABE using lightweight elliptic curve cryptography (ECC) that can secure data at multiple layers still maintaining the lightweight communication. Attribute based encryption is basically a type of public key crypto-system in which the secret sharing key and ciphertext are followed upon some particular attributes. Using lightweight ECC in KP-ABE (key-policy based ABE) or CP-ABE (ciphertext-policy based ABE) can ensure a secure communication between low end devices. Therefore, in this work, we designed and modified a secure version of MQTT that maintains secure communication in IoT sensor networks. A basic architecture of secure MQTT is presented in Figure 1.

The primary objectives of this work are as follows:

- To analyze use of cryptographic approaches in different communication protocols in IoT ecosystem
- To propose a secure version of lightweight MQTT protocol for wireless sensor networks
- To develop a multi-tier authentication system to ensure data privacy in IoT ecosystem

The rest of the paper is organized as follows: Section II presents detailed literature review on security prospects in IoT with analyzing scopes and limitations. Section III discusses the overall methodology of our proposed multi-tier security system with providing necessary flow diagrams. Then Section IV proves the validity of our proposed base. Finally, Section V looks into the applications, scopes, limitations and future directions of the work before coming to a conclusion.

978-1-7281-1325-8/18/$31.00 © 2018 IEEE

II. Literature Overview

A. Background

The use of elliptic curve cryptography (ECC) is nothing new in IoT paradigm to secure data through lightweight communication. It can be used for node authentication and trend says ECC will replace all other public key cryptographic approaches as it is highly efficient with shorter keys [7]. Mutual node authentication is applicable and important for smart home systems [8], healthcare environments [9] and other IoT arenas that requires RFID implant systems based wireless sensor networks [10]. As the numbers of ad-hoc networks are increasing day by day, symmetric key cryptography is no longer supportive and among the asymmetric key algorithms, ECC is best fitted for securing communication in IoT.

There are varieties of lightweight protocols for maintaining communications between low-end devices. But among all others including AMQP and CoAP, MQTT is more popular for device to device (D2D) communications. Therefore, it only needs to secure the protocol using particular authentication and ciphering method [11]. Some works use hash functions as a helping hand for mutual authentication just like [12] did in order to ensure security against the masquerade, stolenverifier, replay, and guessing attacks. In [13], authors used SHA-1 and feature extraction along with ECC to ensure secure mutual authentication.

Some research works have been performed to secure other protocols as well just like [14] utilized ECC to secure the CoAP. But CoAP still being more resource-friendly, in case of message oriented approach MQTT is better [15]. Therefore, another architecture on secure MQTT was published by [16] and they implemented both the RSA and ECC to validate the effectiveness of proposed protocol.

In case of discussing the usage and utilization of CP/KP ABE in securing MQTT protocols, only a few works have been published so far. [17] proposed a secure framework for lightweight devices using CP-ABE with constant sized key and ECC. [11] thereby presented performance of ABE encryption while securing MQTT for sensor networks. They also discussed the feasibility of SMQTT and SMQTT-SN protocols and other security aspects related to D2D communication. And to be named another recent approach named tokenization proposed in [18] is not yet developed for maintaining communication. It just provides data security if implemented in fog or cloud.

Therefore, studying all other works, we decided to focus on secure communication for lightweight devices that are connected to each other and is maintained through server-client processes as well.

III. Methodology

A. Multi-tier approach

Considering IoT vulnerabilities and security design challenges, we proposed a flexible multi-factor authentication system with an encrypted extra secure layer between sensory devices and the cloud. There are primarily three tiers in our proposed security system. IoT ecosystem thus require to address the security concerns at their own layers. The system block diagram is shown in Figure 2.

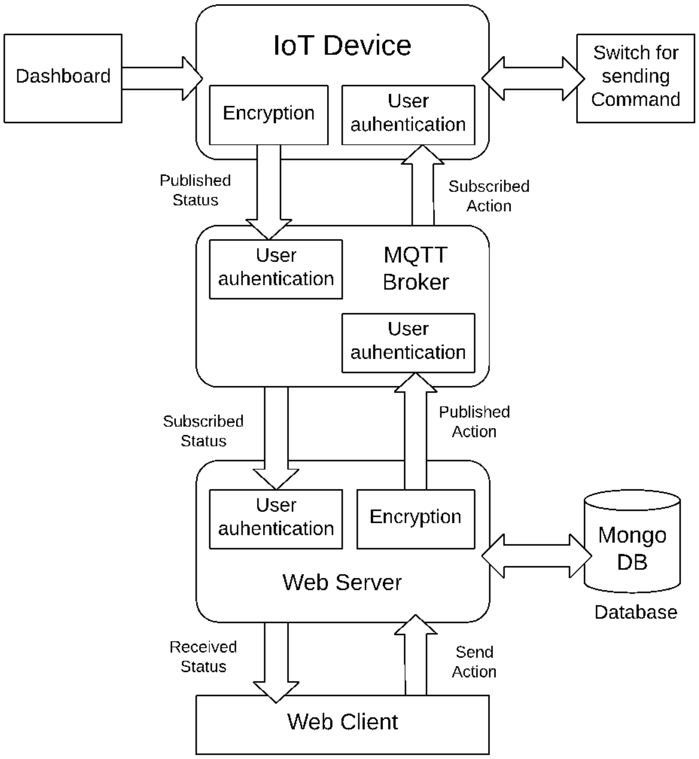


Fig. 2. System block diagram

*1) **Devices/Gateways tier:**:* Hackers often pretends as the authenticated one and try to manipulate using malicious commands. This tier primarily protects the ecosystem against attacks that involves listening to sensitive data being sent from the devices as well. Other considered Security concerns relate to mutual authentication, certificate provision and verification, message encryption, secure booting, firmware updates and patches etc.

*2) **Network/Transport tier:**:* Protects against fabrication and masquerade attack so that device data is able to maintain privacy and integrity of data. The considered Security concerns in this tier relates to device authentication and authorization, application programming interface security, and security concerning message transport.

*3) **Applications tier:**:* Protect against the misuse and fabrication of data residing in applications running in the application tier with a view to ensuring data theft protection. The considered Security concerns relate to application security, secured API call, messages encryption and decryption, message payload verification etc. The application layer is typically more vulnerable rather than other layers requiring further attention and security measurements for local and cloud-based applications, mobile applications and other analytical API based applications as well.

B. System Architecture

A general prototype is developed as follows- at first system initialize itself, configures GPRS, registers to MQTT broker and performs key management operation. After that a device continuously scans the system for sensing bulb, cable & electricity status, encrypts status using ECC algorithm and publish these encrypted status to MQTT broker as a message. The web server wants to keep an up to date database of system status so that it creates a database service. Web server subscribes to MQTT topics. The web service will then be notified when a new status is received from device via MQTT broker. Web server receives authenticated status message from broker, decrypts received status and dumps into MongoDB database. Similarly web client can take action on system by sending command to server; server receives the command from authenticated client and encrypt the command thereby. Then server publish a MQTT message to broker including clients action command, device is subscribed to get message from web server via broker and it receives authenticated message, decrypts message, executes action and finally return action result and systems current status to web client via MQTT broker and web server. An overall in-detailed system architecture is shown in Figure 3.

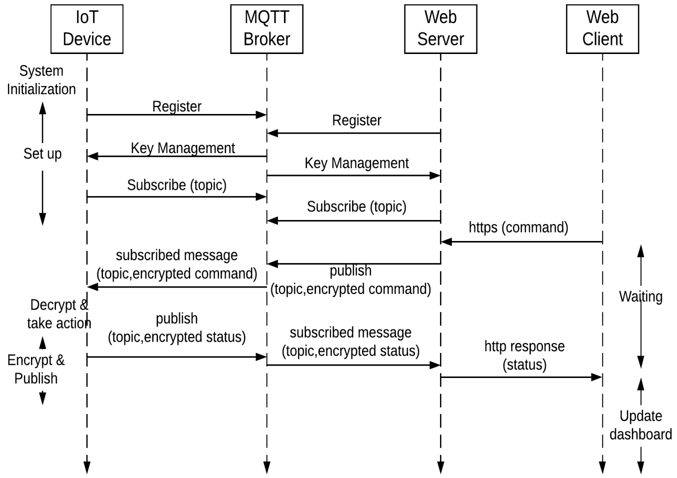


Fig. 3. System Architecture

C. Secure version of MQTT protocol

MQTT provides a lightweight yet secure communication protocol for IoT ecosystem, heterogeneous in nature. The proposed Secure MQTT protocol is powered with lightweight encryption feature using ECC. ECC is a public key crypto-system same as RSA requiring one key to encrypt and another one to decrypt. ECC is calculated based on the mathematics of elliptic curves utilizing the co-ordinate of points on an elliptic curve to perform encryption and decryption. The primary advantage of using ECC is it uses a much shorter key length than other public key crypto-systems to provide an equivalent level of security. In addition to that, the level of security ECC provides, increases faster with key size than for integer based discrete logarithm or RSA based crypto-systems. ECC provides a faster implementation as well and requires less bandwidth and power. Therefore, ECC is important for resource constraint systems where computing power, memory and battery life are limited.

D. Algorithms

Algorithms of work flow in web client side, web server side and device end side are presented in Algorithm 1, Algorithm 2 and Algorithm 3 respectively.

Algorithm 1: Algorithm at Web client side

```
System Initialization;
while TRUE do
    Send https request to server for current device status;
    if server response with device status then
        Update dashboard;
    end
    /* from dashboard                                  */
    if user press on device ON/OFF switch then
        Send https(action command) request to server;
        Wait for feedback;
        Receive feedback;
        Update dashboard ;
    end
    Wait 10 seconds
end
```

Algorithm 2: Algorithm at web server side

```
System Initialization;
Register, share keys & subscribe topic to MQTT broker;
while TRUE do
    if Receive action command request from client then
        Dump action command into MongoDB and set
          $action\_flag = 1$ ;
    end
    Read action flag from DB;
    if $action\_flag = 1$ then
        Encrypt message(topic & command) using ECC;
        while Status not received do
            Publish Encrypted message to MQTT broker;
            Wait for status;
        end
        if subscribed event triggered then
            Receive encrypted message(status) from the
              device via MQTT;
            Decrypt message;
            Dump status into MongoDB and set
              $action\_flag = 0$;
            Send feedback to web client ;
        end
    end
end
```

Algorithm 3: Algorithm at Device side

```
System Initialization;
Read backup status from EEPROM and update system ;
while TRUE do
    Send AT command to GSM module;
    while GSM is not Ready do
        Wait 100ms ;
        Send AT command to GSM module;
    end
    else
        Configure GPRS for internet access;
        Register, share keys and subscribe topic to
          MQTT broker;
    end
    Scan Systems hardware;
    if any changed found then
        Encrypt message of current status using ECC;
        Send ACK: Publish(topic, encrypted message);
    end
    if received message from MQTT broker then
        Decrypt message(Command) using ECC;
        Turn ON/OFF device based on command;
        Encrypt message of current status using ECC;
        Send ACK: Publish(topic, encrypted message);
    end
    else
        Need Something at least
    end
end
```

IV. Implementation and Results

A. Implementation

1) Hardware Components: In this work, we used Arduino Uno which is a microcontroller board based on the ATmega328P. Arduino is an open-source platform used for constructing and programming of electronics. ECC algorithm is running on this MCU. MCU read the light status, electricity status and command to turn ON/OFF light. 5v Power is used to power up the system. SIM900A GSM/GPRS Module is used for internet connectivity to send or receive data to/from web server. Relay is used to switch light ON/OFF. Relays are simple switches which are operated both electrically and mechanically. ACS712 current sensor is used for sensing the condition of the light/cable, measure current for measuring consumed power. Battery is used to backup power of the system.

2) Software Components: Here, Node Js Server is maintained as the web server and MongoDB for managing the database. Finally we used Eclipse Mosquitto which is an open source (EPL/EDL licensed) message broker that implements the MQTT protocol versions 3.1 and 3.1.1.

B. Result and Performance

RSA is widely used as public key crypto-system in secure communication including SSL. As computation power is being increased day by day, it requires longer key size (in term of RSA- from 1024 to 2048 bits) to maintain security strength. ECC is another public key crypto-system, yet lightweight. The advantage of using ECC over RSA is to provide same level of security strength at much smaller key sizes which is provided in Table I.

According to Table I and Figure 4, we see that ECC provides the same cryptographic strength as an RSA-based public key crypto-system with notable smaller key sizes at greater length. For example, a 224 bit ECC key is equivalent to RSA 2048 bit keys that is a notably difference, yet ECC is comparatively much lightweight than RSA. As most of the secure symmetric key crypto-systems such as AES, utilized in TLS use at least 128 bit keys, it is recommended to use greater length public key crypto-system to maintain necessary security measurements[19].

TABLE I
Ratio in Key-size, Security bits and ratio of cost measured for EEC in comparison with RSA/DSA [20]

Key Size		Ratio of	Security	Ratio
RSA/DSA	ECC	Key Size	level (bits)	of Cost
1024	160	7:1	80	3:1
2048	224	10:1	112	6:1
3072	256	12:1	128	10:1
7680	384	20:1	192	32:1
15360	521	30:1	256	64:1

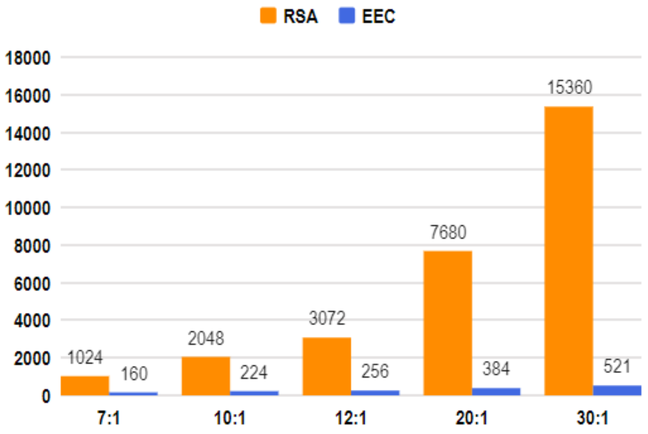
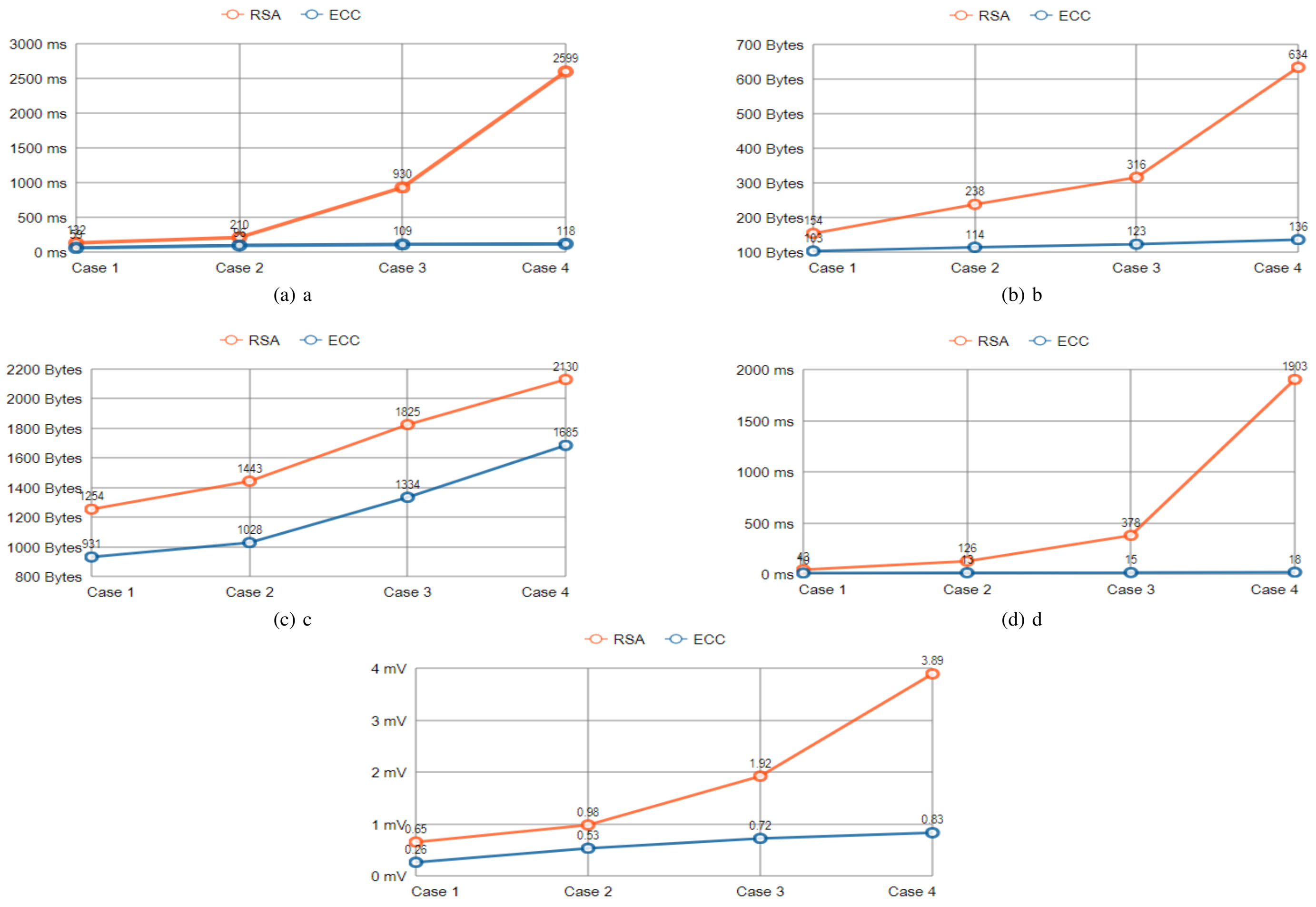


Fig. 4. Key Comparison between RSA/DSA and EEC

After implementation we found that ECC works comparatively more efficient in lower end devices. The comparison of ECC and RSA that is tested in UNO/ATmega328P @16MHz is presented in Table II and Figure 5.

TABLE II
ECC Vs RSA performance on Arduino UNO/ATmega328P @16MHz

Security equivalence	Key Size (bits)		Key Generation Time (ms)		Data memory /RAM (Bytes)		Code memory /ROM (Bytes)		ECC Encrypt/Decrypt Time (ms)		Battery Drain (mV)	
	ECC	RSA	ECC	RSA	ECC	RSA	ECC	RSA	ECC	RSA	ECC	RSA
Case 1	100	512	59	132	103	154	931	1254	10	43	0.26	0.65
Case 2	130	764	96	210	114	238	1028	1443	13	126	0.53	0.98
Case 3	160	1024	109	930	123	316	1334	1825	15	378	0.72	1.92
Case 4	256	2048	118	2599	136	634	1685	2130	18	1903	0.83	3.89



(a) a

(b) b

(c) c

(d) d

(e) e

Fig. 5. Comparison between EEC and RSA for equivalent security based on- (a) Key generation time, (b) data memory, (c) code memory, (d) encryption/decryption time, (e) battery drainage

V. Discussion

A. Significance and Application

The very first thing to be addressed properly once a device is given connectivity to the internet is its security. Also, we need to think about the performance in energy and power. Therefore, our research focused on avoiding such security vulnerabilities by utilizing lightweight crypto-system for lower end devices. In order to remove the extra overhead due to high memory and computing power required by the traditional encryption algorithm, we utilized ECC to embed a lightweight and faster encryption. Adoption of this technique has been made possible due to its minimum public key size which is 160 bit compared to 1024 bit of RSA and DSA with still maintaining the equivalent level of security. The result shows that only 1.16 KB of Dynamic Ram is being used for running the encryption which is a great improvement in terms of the memory consumption. So this technique proves to be very handy for IoT devices with memory and computing power constraints.

Furthermore, to fortify the existing less secure MQTT communication protocol used to communicate from device to server, we have implemented SSL/TLS security in the MQTT broker end. This provides end to end encryption for both the client and server and ensures an extra layer of security to our current security layout.

Another important research outcome is- the security alert in the event of any theft or tampering of the device. Whenever such attempt of device tampering arises, a sensor will instantly detect and notify the user of its current status whether the device is disconnected or if the device is consuming electricity. Device can also be remotely controlled and monitored using this method from anywhere. This feature ensures a device security at the installation end which is often difficult to keep track of, eliminating the need of direct human supervision.

B. Scope and Limitations

We focused on the utilization of ECC as it performs better where computing power, memory and battery life are limited. Many smart card, cell phone, IoT and Bitcoin businesses have already adopted ECC. The only limitation is- ECC requires some agreement on which type of curve and curve parameters to use; where support for ECC, especially more modern curves, is lacking from many libraries. RSA is much easier to understand than ECC (and a better understanding aids the security of protocols and implementations); and RSA has been much better researched than ECC, e.g. with regards to side channel attacks.

C. Future Work

We are going to develop a crypto chip where ECC algorithm will be embedded. This will be a dedicated chip for encryption and decryption and can be configured for lightweight lower end sensor devices.

VI. Conclusion

In order to maintain secure communication between heterogeneous IoT ecosystem, device to device communication requires further security along with lightweight cryptosystems. Sensor oriented devices in IoT ecosystem are resource constraint with limited computation power and memory capacity that makes it difficult to apply heavier cryptosystems. Being the best protocol for transmitting light weight data (sensory data), securing MQTT requires further attention as systems using MQTT consume less power, has great efficiency of transmission and can utilize system bandwidth efficiently. We performed a performance measure that shows that using ECC with CP/KP-ABE provides a comparatively far more better lightweight communication for low-end devices than using traditional cryptographic approaches.

Acknowledgement

The research work was fully supported by Ministry of Information and Communication Technology (MoICT), Bangladesh. Partial support has been provided as well from Green University of Bangladesh research fund.

References

[1] D. Mourtzis, E. Vlachou, and N. Milas, "Industrial big data as a result of iot adoption in manufacturing," *Procedia CIRP*, vol. 55, pp. 290–295, 2016.

[2] M. Ozger, O. Cetinkaya, and O. B. Akan, "Energy harvesting cognitive radio networking for iot-enabled smart grid," *Mobile Networks and Applications*, pp. 1–11, 2017.

[3] S. Roy, A. Rahman, M. Helal, M. S. Kaiser, and Z. I. Chowdhury, "Low cost rf based online patient monitoring using web and mobile applications," in *Informatics, Electronics and Vision (ICIEV), 2016 5th International Conference on*. IEEE, 2016, pp. 869–874.

[4] J. Shenoy and Y. Pingle, "Iot in agriculture," in *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on*. IEEE, 2016, pp. 1456–1458.

[5] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *Journal of Network and Computer Applications*, vol. 40, pp. 325–344, 2014.

[6] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "Mqtt-sa publish/subscribe protocol for wireless sensor networks," in *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on*. IEEE, 2008, pp. 791–798.

[7] S. Guicheng and Y. Zhen, "Application of elliptic curve cryptography in node authentication of internet of things," in *Intelligent Information Hiding and Multimedia Signal Processing, 2013 Ninth International Conference on*. IEEE, 2013, pp. 452–455.

[8] F. K. Santoso and N. C. Vun, "Securing iot for smart home system," in *Consumer Electronics (ISCE), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 1–2.

[9] D. He and S. Zeadally, "An analysis of rfid authentication schemes for internet of things in healthcare environment using elliptic curve cryptography," *IEEE internet of things journal*, vol. 2, no. 1, pp. 72–83, 2015.

[10] S. R. Moosavi, E. Nigussie, S. Virtanen, and J. Isoaho, "An elliptic curve-based mutual authentication scheme for rfid implant systems," *Procedia Computer Science*, vol. 32, pp. 198–206, 2014.

[11] M. Singh, M. Rajan, V. Shivraj, and P. Balamuralidhar, "Secure mqtt for internet of things (iot)," in *Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on*. IEEE, 2015, pp. 746–751.

[12] T.-H. Chen and W.-K. Shih, "A robust mutual authentication protocol for wireless sensor networks," *ETRI journal*, vol. 32, no. 5, pp. 704–712, 2010.

[13] G. Zhao, X. Si, J. Wang, X. Long, and T. Hu, "A novel mutual authentication scheme for internet of things," in *Modelling, Identification and Control (ICMIC), Proceedings of 2011 International Conference on*. IEEE, 2011, pp. 563–566.

[14] A. F. Skarmeta, J. L. Hernandez-Ramos, and M. V. Moreno, "A decentralized approach for security and privacy challenges in the internet of things," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 67–72.

[15] K. Fysarakis, I. Askoxylakis, O. Soultatos, I. Papaefstathiou, C. Manifavas, and V. Katos, "Which iot protocol? comparing standardized approaches over a common m2m application," in *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE, 2016, pp. 1–7.

[16] 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 *Systems of Collaboration (SysCo), International Conference on*. IEEE, 2016, pp. 1–6.

[17] V. Odelu, A. K. Das, and A. Goswami, "An efficient cp-abe with constant size secret keys using ecc for lightweight devices." *IACR Cryptology ePrint Archive*, vol. 2015, p. 841, 2015.

[18] S. Roy, A. R. Shovon, and M. Whaiduzzaman, "Combined approach of tokenization and mining to secure and optimize big data in cloud storage," in *Humanitarian Technology Conference (R10-HTC), 2017 IEEE Region 10*. IEEE, 2017, pp. 83–88.

[19] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "Recommendation for key management part 1: General (revision 3)," *NIST special publication*, vol. 800, no. 57, pp. 1–147, 2012.

[20] M. Bafandehkar, S. M. Yasin, R. Mahmod, and Z. M. Hanapi, "Comparison of ecc and rsa algorithm in resource constrained devices," in *IT Convergence and Security (ICITCS), 2013 International Conference on*. IEEE, 2013, pp. 1–3.