2016 Symposium on Colossal Data Analysis and Networking (CDAN)

MQTT Based Secured Home Automation System

Yuvraj Upadhyay
M.Tech-VLSI & Embedded System Design
Vel Tech Dr.RR & Dr.SR Technical University
Chennai Tamil Nadu, India
Email: upadhyayyuvraj@gmail.com

Mr.Amol Borole
Qualex Systems Pvt Ltd.
Pune, India
Email: aborole@gmail.com

Mr.D.Dileepan
Vel Tech Dr.RR & Dr.SR Technical University
Chennai Tamil Nadu, India.
Email: dileepan@veltechuniv.edu.in

Abstract—**In this paper we discuss and create a MQTT based Secured home automation system, by using mentioned sensors and using Raspberry pi B+ model as the network gateway, here we have implemented MQTT Protocol for transferring & receiving sensor data and finally getting access to those sensor data, also we have implemented ACL (access control list) to provide encryption method for the data and finally monitoring those data on webpage or any network devices. R-pi has been used as a gateway or the main server in the whole system, which has various sensor connected to it via wired or wireless communication.**

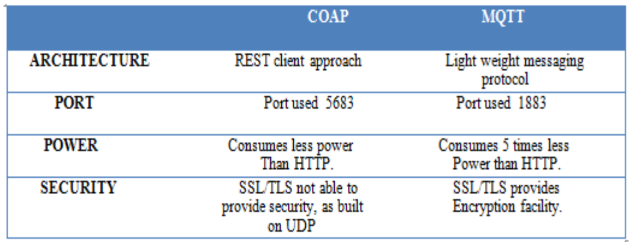
Keywords-ACL, MQTT, COAP, MQTT-SN, Raspberry-pi

I. Introduction

Today IoT has become an good platform which is used in various areas be it in industrial automation or home automation or medical purpose, connecting various sensor for monitoring various device activities and monitoring those data from anywhere in the region so to provide more flexibility to the user .However whenever IoT is referred security is the main concern that needs to be given to the data. With advancement in IoT field various methods and protocols have been introduced in order to impart better safe data transmission like MQTT, MQTT-SN[5] or COAP. MQTT is message queuing telemetry transport light weight messaging protocol mainly IOT connectivity protocol, working on default TCP/IP port 1883 MQTT are of various type hive-mq ,mosquitto,paho mqtt, however here we have used MQTT-mosquitto Further adding, even though with the implementation of these protocols security remains a main concern for that various methods are implemented be it ACL or hash algorithm with SSL support for data encryption and security .These data encryption methods are important because it protects our privacy and data from being misused or hampered .The encryption method best works when its independent. The best part is it protects from bulk surveillance.

In the given system Raspberry-pi is used as a gateway which has sensors connected to it using through wired and ZigBee protocol, e.g. DHT11 and DS18B20 to monitor temperature and humidity of different sections in which the sensors are placed. The raspberry pi is then configured with MQTT and lastly with ACL which is used for encryption of data. With these the raspberry then periodically collects the sensors data, which is accessed by the user using webpage or through any other device .Here we have used the webpage and an app to display and access our data, also created a data base using redis backend mysql to store data. Other database option one can use is MYSQL,SQLite3 database ,for this one should need to configure it by properly giving correct host and port number, moreover we have also tried to show the whole process by using an platform of gadgetkeeper, thinspeak.

TABLE I
COAP VS MQTT COMPARISON



	COAP	MQTT
ARCHITECTURE	REST client approach	Light weight messaging protocol
PORT	Port used 5683	Port used 1883
POWER	Consumes less power Than HTTP.	Consumes 5 times less Power than HTTP.
SECURITY	SSL/TLS not able to provide security, as built on UDP	SSL/TLS provides Encryption facility.

II. Design and Implementation

In this proposed architecture we have created a home automation system with MQTT and ACL, on brief note, various sensors are connected to it ,then installation of MQTT and ACL done and using this the sensor data is sent to the user the whole project is divided in three sections sensor node, MQTT-Mosquitto, ACL-access control list .In the first part the raspberry pi which is used as an gateway is connected to sensors via both wired and wireless connection[3], here we have used DHT11 temperature and humidity sensor and DS18B20 sensor .The first phase includes the testing of the sensors and raspberry pi .Second phase includes the implementation of MQTT

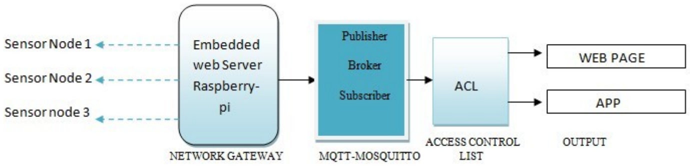


Fig. 1. SYSTEM ARCHITECTURE

Mosquitto, firstly the mosquitto is installed and it's tested by exchange of messages between publisher and subscriber,

978-1-5090-0669-4/16/$31.00 ©2016 IEEE

In first terminal the subscriber is run using the command mosquitto sub -t topic and in other terminal publisher command is run to display the data to user using mosquitto pub -t topic -m message.

The concept of mqtt is simple ,it has three parts the publisher ,broker and subscriber .Whenever the data arrives at publisher the data topic is sent to broker which stores it ,in manner we can say broker is kind of filter which only filters or sends those data/messages further which are requested .Later the subscriber subscribes to any topic or data that it wants to see or get access ,upon receiving this request the broker alerts the publisher and the publisher responds it to the request there by sending the topic or the data of interest requested and hereby finally user sees the data. So whenever the sensor data is sensed by the sensor its send to the R-Pi device where we subscribe to the topic which we have created to view the temperature using the above method mentioned. At final stage implementation and use of ACL or access control list is made ,this property is applied in order to provide the safety to the data that is being transferred or in a simple way we can say encryption method is imparted over the data .This stage includes the database management ,introduction of PSK and logging methods by creating username and password for accessing the data ,in short the ACL strengthens the whole MQTT system, hence it becomes mandatory to use it .Thus with this we have made the ACL communicate with the MQTT protocol and get the data using same publish and subscribe method .Then the data is displayed using the webpage also we used the MQTT app to publish the data by giving the default port used by MQTT and subscribing the topic name.

III. MQTT ARCHITECTURE

MQTT stands for message queuing telemetry transport. Unlike COAP, TCP, UDP, it is used because MQTT specializes in low-bandwidth, high-latency environments; it is an ideal protocol for machine-to-machine (M2M) communication. Basically there are three components in MQTT publisher, subscriber and broker. Here the process of receiving and publishing the data is very much secure and accurate. Whenever the user wants to check or go through any data it sends the request to broker and upon receiving the request it sends to the publisher, it responds to the requests and sends the data that is requested by the subscriber and hence publishes the data, in overall process the communication is secure and upto the topic of interest. MQTT broker acts like a filter allowing only those data which are requested thereby saving the flow of ambiguous data.

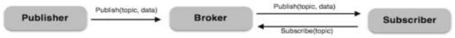


Fig. 2. MQTT Process

IV.MQTT METHODS AND TESTING

Now we can go ahead and install Mosquitto proper. There are three packages:

- mosquitto is the MQTT broker (i.e. server)
- mosquitto-clients are the command-line clients, which I recommend we install
- python-mosquitto are the Python bindings, which I also think we should install

The next step is to test the mosquitto and the publisher

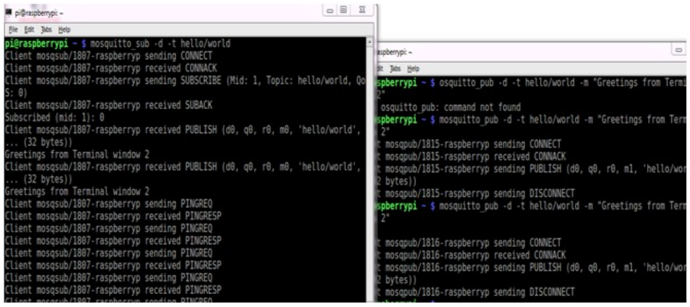
In separate terminal windows do the following:
Start the broker: mosquitto

i>Start the command line subscriber:
mosquitto_sub -v -t 'test/topic'

ii>Publish test message with the command line publisher:
mosquitto_pub -t 'test/topic' -m 'helloWorld'

iii>As well as seeing both the subscriber and publisher connection messages in the broker terminal the following should be printed in the subscriber terminal:

Test/topichelloWorld



```
pi@raspberrypi ~ $ mosquitto_sub -d -t hello/world
Client mosqsub/1807-raspberryp sending CONNECT
Client mosqsub/1807-raspberryp received CONNACK
Client mosqsub/1807-raspberryp sending SUBSCRIBE (Mid: 1, Topic: hello/world, Qo
S: 0)
Client mosqsub/1807-raspberryp received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/1807-raspberryp received PUBLISH (d0, q0, r0, m0, 'hello/world',
... (32 bytes))
Greetings from Terminal window 2
Client mosqsub/1807-raspberryp received PUBLISH (d0, q0, r0, m0, 'hello/world',
... (32 bytes))
Greetings from Terminal window 2
Client mosqsub/1807-raspberryp sending PINGREQ
Client mosqsub/1807-raspberryp received PINGRESP
Client mosqsub/1807-raspberryp sending PINGREQ
Client mosqsub/1807-raspberryp received PINGRESP
Client mosqsub/1807-raspberryp sending PINGREQ
Client mosqsub/1807-raspberryp received PINGRESP
Client mosqsub/1807-raspberryp sending PINGREQ
Client mosqsub/1807-raspberryp received PINGRESP
```

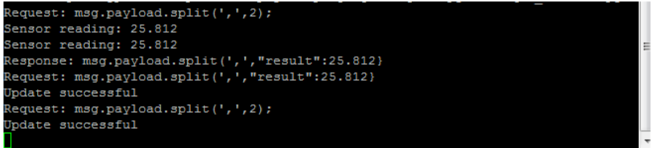
```
spberrypi ~ $ osquitto_pub -d -t hello/world -m "Greetings from Termi
2"
 osquitto_pub: command not found
spberrypi ~ $ mosquitto_pub -d -t hello/world -m "Greetings from Term
 2"
t mosqpub/1815-raspberryp sending CONNECT
t mosqpub/1815-raspberryp received CONNACK
t mosqpub/1815-raspberryp sending PUBLISH (d0, q0, r0, m1, 'hello/wor
2 bytes))
t mosqpub/1815-raspberryp sending DISCONNECT
spberrypi ~ $ mosquitto_pub -d -t hello/world -m "Greetings from Term
 2"

t mosqpub/1816-raspberryp sending CONNECT
t mosqpub/1816-raspberryp received CONNACK
t mosqpub/1816-raspberryp sending PUBLISH (d0, q0, r0, m1, 'hello/wor
2 bytes))
t mosqpub/1816-raspberryp sending DISCONNECT
```

Fig. 3. MQTT Testing

Similarly if we look for ACL it is a method that if we are planning to strengthen our MQTT service, then access control lists (ACL) are mandatory. Mosquitto broker supports this ACL feature through auth plugging. An access control list (**ACL**), with respect to a computer file system, is a list of permissions attached to an object.

IV. METHODOLOGY

The main requirments of the proposed system is to acquire the data of the sensors and send them by using security protocol to the user whenever the user wishes to check the working or the environment data of its home or other place. The raspberry pi acts as a home gateway server connected to the home network, thereby acting as a mini system having control over the sensors which gives the updates about the home environment from temperature to humidity and also status of the devices. Likewise here we have used DHT11/22 and DS18B20 Sensors for our project. Moreover the data is stored continuously and the advantage is no ambiguous data is stored and its all due to the MQTT and ACL implementation. Also to prevent transmission of any faulty or ambiguous data and to restrict the access to the data to authorized users only the ACL method of encryption is used which provides login facilities secured by ssl. The figure 4 shows the output that we received, when we make a subscription request from the application i.e. android app.



```
Request: msg.payload.split(',',2);
Sensor reading: 25.812
Sensor reading: 25.812
Response: msg.payload.split(',',"result":25.812)
Request: msg.payload.split(',',"result":25.812)
Update successful
Request: msg.payload.split(',',2);
Update successful
```

Fig. 4 Reading

V. RASPBERRY PI HOME GATEWAY

It's an small SOC device, a mini computer with almost all configuration properties to interface with any devices or sensors etc.As mentioned in this project raspberry pi is a gateway to which various sensors are interfaced, on which MQTT and ACL methods are installed and tested. Like here in this paper we have made use of the two sensors DHT11/22 and DS18B20, which are interfaced with the raspberry pi through wired and wireless connection.

A. *FEATURES*

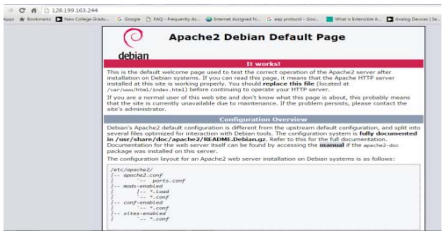
In this project we have used raspberry pi B+ model with 512mb RAM and ARM 7 with 1GHZ processor and 4 USB ports with an Ethernet port, has 40 GPIO pins is operated at 1.8mA and 5v power supply. For OS porting SD card is used with Raspbian as installed OS, it is based on Linux operating system based on debian, during first boot its necessary to activate various advanced options from SSH to wifi enabling options.

B. *SSH*

Its known as secure shell, its main work is to provide remote access of the terminal of raspberry pi to the other system in same network .In order to get the whole raspberry pi setup screen on the other system either we can use XMING or VNC server .Here we have used the XMING below is the image showing putty and xming setup to get raspberry pi desktop

C. *APACHE SERVER*

The apache server is basically used to create the embedded web server. Here we have installed the apache server in order to display the output data on localserver .It can be tested using HTML script.



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|        `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

Fig. 5. APACHE SERVER

VI. SENSORS

In our current project we have made use of the two temperature sensor and also to ensure the performance of the two with the MQTT protocol, one is DHT11/22 and second one is DS18b20.

A. *DHT11/22 Temperature Humidity sensor*

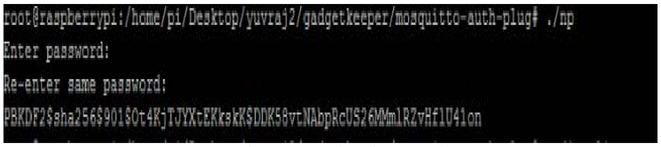
This sensor is used to detect both temperatures as well as humidity .Basically has 4 pins and uses capacitive humidity sensor to measure air and thereby generates a digital signal on the data pin .Generally works on 3 to 5v power supply. The first pin is VDD power supply, second pin is the data pin, third pin is no connection pin and last pin is ground.

B. *DS18B20 Temperature Sensor*

This sensor is also used to measure the temperature, has three pin ,the first pin is the ground ,second pin is the data pin and the third pin is the vdd pin .It simplifies distributed temperature sensing properties with multiple drop capability.

VII. IMPLEMENTING MQTT AND ACL

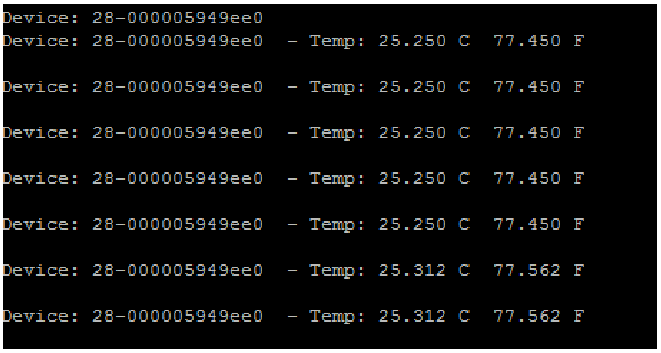
Firstly we showed how we have used the MQTT protocol with the setup that is the sensors and the raspberry-pi, also we have used the ACL method that is acting as an extra security to the whole process. For each data we have successfully created a username and password method in order to get access to the data. Below Figure shows the result, that how we have obtained the ACL implementation result, we achieved this by making use of mosquitto auth plugin, which is a useful for implementing the ACL with MQTT.



```
root@raspberrypi:/home/pi/Desktop/yuvraj2/gadgetkeeper/mosquitto-auth-plug# ./np
Enter password:
Re-enter same password:
PBKDF2$sha256$901$Ot4KjTJYXtEKksKK$DDK58vtNAbpRcUS26MMmlRZvHf1U41on
```

Fig. 6 ACL implementation

The following figure below of the temperature data we obtained after the successful implementation of MQTT with ACL



```
Device: 28-000005949ee0
Device: 28-000005949ee0  - Temp: 25.250 C  77.450 F

Device: 28-000005949ee0  - Temp: 25.250 C  77.450 F

Device: 28-000005949ee0  - Temp: 25.250 C  77.450 F

Device: 28-000005949ee0  - Temp: 25.250 C  77.450 F

Device: 28-000005949ee0  - Temp: 25.250 C  77.450 F

Device: 28-000005949ee0  - Temp: 25.312 C  77.562 F

Device: 28-000005949ee0  - Temp: 25.312 C  77.562 F
```

Fig.7 sensed data

VIII.CONCLUSION

In this given project we have tried to create an home automation and making it more efficient and secure by using MQTT protocol and ACL for the user .The whole setup and mechanism has a better and efficient advantage over the other protocols like COAP,TCP/UDP,HTTP etc ,moreover it also overcomes the home stack protocol of apple .With this proposed system one can use and deploy it in industrial as well as home automation purpose ,as it provides security followed by encryption of data to database management system .Since its not possible to cover each and every issues in the field of IoT but We have tried to overcome as many issues as we can that is faced by previous architectures

ACKNOWLEDGEMENT

I would like to sincerely express my gratitude towards my guides from CDAC, Pune and VELTECH, University for guiding me in such a project without whom it wouldn’t have been possible.

REFERENCES

[1] QoS scheme for multimedia multicast communications over wireless mesh networks, M. Iqbal, X. Wang, S. Li, and T. Ellis,Faculty of Computing, Information Systems and Mathematics, Kingston University, Kingston upon Thames, Surrey KT1 2EE, UK School of Engineering, Swansea University, Swansea, UK ,E-mail: m.iqbal@wirelesslive.net IET Commun vol. 4, no. 11, pp. 13121324, Jul. 2011.

[2] Smart home implementation based on internet and wi-fi technology,”YAN Wenbo ,WANG Quanyu, GAO Zhenwei Control Conference (CCC), 2015 34th Chinese

[3] Wi-Fi-based Hierarchical Wireless Networked Control Systems, Esraa A. Makled, Hassan H. Halawa, Ramez M. Daoud, Hassanein H. Amer Electronics and Communications Engineering Department American University in Cairo (AUC) Cairo, Egypt, IEEE 2015

[4] Raspberry Pi as a Wireless Sensor Node: Performances and Constraints, Vladimir Vujovic and Mirjana Maksimovic, Faculty of Electrical Engineering, East Sarajevo, Bosnia and Herzegovina, IEEE 2014

[5] MQTT for Sensor Networks (MQTT-SN) Protocol Specification , Andy Stanford-Clark and Hong Linh Truong , Andy Stanford-Clark and Hong Linh Truong (andysc@uk.ibm.com, hlt@zurich.ibm.com) November 14, 2013

[6] MQTT-S A publish/subscribe protocol for Wireless Sensor Networks, Hunkeler, U. ; IBM Zurich Res. Lab., Zurich ; Hong Linh Truong ; Stanford-Clark, A , Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference