International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1S5, December 2019

Security Enhancement for Devices using REST-API, Middleware & IoT Gateway

Shaik. Jakeer Hussain, Gudapati Ramyasri

Abstract: ***Internet of things plays the major innovative role in the enhancement and optimization of habitual behavior by the collaborative usage of smart objects and smart sensors. One of the major challenges that is being faced is secure interconnection of iot devices, sensors, actuators to the cloud. As most of the IoT devices and cloud usage is done using the third party, it is required to provide IoT security such that the attackers cant disturb the communication path through the devies and also to provide secure data transmission from devices to cloud. This paper mainly displays the use of middleware along with Gateway and REST API Technologies for the secure interface between the devices, sensors and storage applications.***

Keywords: Internet of Things, security, Gateway, REST API, Middleware.

I. INTRODUCTION

Internet of Things (IoT) Playing a major role in connecting the things physically and virtually. This interconnection of the physical devices to the digital network, data can be collected and analyzed from anywhere. So, the major challenge that is being faced is privacy, security issues by these devices and also formation of secure communication path between devices. Now a days, most of the IoT devices are linked up with default cloud server and particular applications. So, the user can't change to more secured services [14].

The security services that are being used are related to the third party. So, there are chances of misusage and miss transmission of data. Even the devices that are being connected are not given unique identity. The attackers on Iot devices not only steel data from server but also cause attacks on nearby systems. Devices in IoT can transfer the data based on the registration to an adequate authorized app in a personal area network but all the devices in IoT don't have the feature of authorized transmission of data. For this device environment and resource registration is done and the data is stored in the server in an encrypted way.

Even though REST API offers best and HTTP protocol is used in real time applications. In this architecture for constrained network formation and transmission of data between client servers MQTT is used. MQTT broker acts as mediator for the formation of request response model using MQTT publication and subscription [2].

Manuscript published on 30 December 2019.
* Correspondence Author (s)
Shaik.Jakeer Hussain Professor, School of Electronics, VFSTR, Guntur jk.shaik@gmail.com
Gudapati Ramyasri, Research Scholar, School of Electronics, VFSTR, Guntur, ramyasri.gudapati@gmail.com

II. MESSAGE QUEUE TELEMETRY TRANSPORT

Message queue telemetry transport is a light weight protocol which uses publisher-subscriber model with a centralized broker for device to device communication. MQTT is mainly for IOT devices for light weight communication path with 2 byte header. MQTT mainly contains MQTT client and MQTT Server. MQTT client is nothing but publisher or subscriber which is a device to send or receive data. MQTT Server is nothing but broker which collects published data and transmits data to related subscribers.
Before the MQTT client and the server can exchange the data and packets, the synchronization between the client and the server is made. This is done based on the formation of Topics. The MQTT software first should be authorized by using openssl and ca key and authorization. So that data transmission is done securely. MQTT is mainly done in three steps

i. Establishing a Connection MQTT Client and Server

The Connection between the mqtt client ans the server is mainly formed using CONNECT Packet transmission from the client to server. The CONNECT packet consists of the unique MQTT client identifier, flags representing the status and the type of protocol that is being used. The CONNACK packet will be sent for the acknowledgement specifying the status of connection between the client and the server.

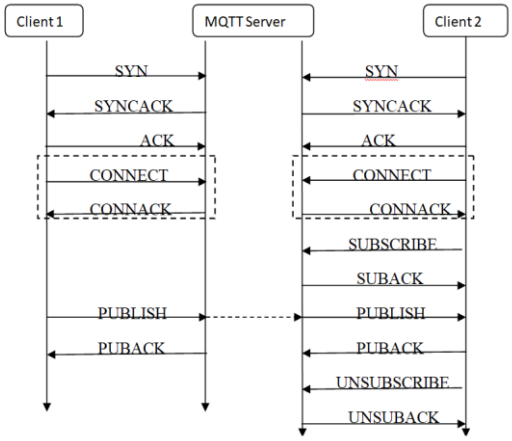

Fig.1. Request Response model using MQTT

ii. Maintaining the connection

After certain period of time, the connection between the client and the server will be terminated. To keep that connection alive for more period of time PINGREQ is sent from client to server and
PINGRES packet as acknowledgement is sent back.

iii. Terminating the connection

To end up the reception of data from broker, UNSUBSCRIBE packet is sent.
In response UNSUB packet is sent back which automatically stops the reception of data from server.

In MQTT, subscriber sends the subscription request to the broker along with topic, userid, authorized username and password. The response message consists of a status code and a payload[5]. Publisher sends the requested data to the broker along with topic, user id, authorized username and password. So that the transmission of data is done in encrypted form to a authorized user. Broker only sends the data to requested subscriber. The centralized broker acts as a media for communication between all subscribers and publishers. The MQTT is designed so that the subject for the response message may be derived from the request mechanically.

III. MIDDLEWARE

IoT Middleware is software which acts a media for interaction between the two devices [9]. There is no any Middleware which is general for all type of applications. So, Middleware should be designed for specific applications with limited time taking and power consumption without causing any security problems and increasing privacy [1].

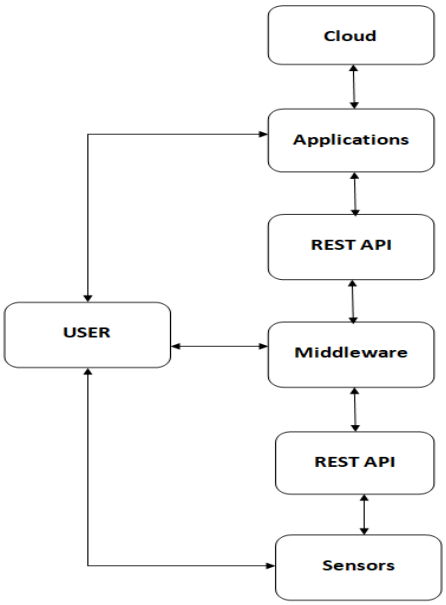

Fig.2. IoT Middleware

Fig.2. shows the basic IoT middleware platform connecting things to cloud for storage of data. Even though both are using same API, Middleware acts a media of communication.

Middleware is mainly acts as translator between two different applications. Any two applications won't use API designed in same language. There many middleware platforms that are developed still now[10]. Some middleware networks are developed based on enabling technologies of applications and device management. Some middleware platforms are developed based on the device management and enabling software. Some are based on the enterprise and M2M iiot based [1].

Taking personal Hub as reference for enabling software, I had designed home automation network with middleware formation using Personal Network Hub. Personal Network Hub is used to generate the user id for each application which is used identity for user [4].

IV. PROPOSED WORK

In Internet of things if every device is connected to the internet means each device power consumption and energy utilization will be more. The circuit design also will be complex. Unique identity of each device and direct transmission of data from Middleware to the device also will be complex. So, gateway is used as media to interface devices to the internet. Through a single gateway all devices and middleware are interfaced with each other[8].

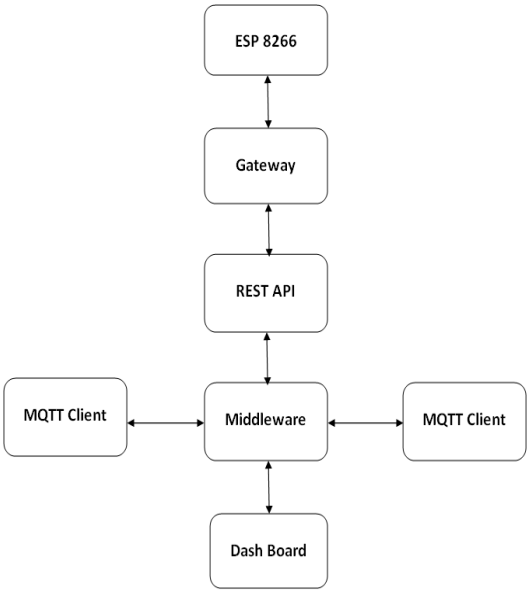

Fig.3. IoT Middleware Architecture

Fig.3 shows the proposed model of communication through the middleware. Middleware acts as a media to transfer the data from gateway to the registered application (either it's an app or Cloud). The status of the Middleware is displayed in dashboard. Dashboard represents the status of registration, identity, and transmission [15].

Device manufacturers itself represents unique identity for a device. Using these identification numbers along with unique token registration of the devices will be done and communication path is formed between the gateway and device. The user identification is done using the identity provided by the Google account or open Id providers that are available online.

Based on registration of the devices, applications they can transmit the data but know the identity of each other than token exchange. Even devices and applications don't know the user id also. They just transmit the data based on the token provided and the validity of data. Device data that is getting transmitted is encrypted and transmitted from the node itself[6].

The registered data will be stored in data server and for transmission of data gateway will check the token in the server and based on proper authentication of both publisher and subscriber it will transmit the data. Gateway also checks the topic and username password and identity provided for both publisher and subscriber should be same. Based on that proper authentication it will recognize the identity and transmits the data [7].

Middleware is designed to register the unique identity of each device, registration of each device and transmission of data. Using REST API the application and Database will be interfaced to the middleware and management of data storage is done. Based on Authorization and Authentication of the users using unique token and registration ID data is transmitted to the application and can be viewed in the dashboard.

The proposed model of registration for devices, user and applications is shown in the Fig.4. Here Device registration is done based on the id provided by manufacturer of device under the proper authorization of user. User registration is done based on user id provided by any of the authorized account [5]. Once the device registration is over, the user can authorize to any application for the transmission of data detected from the sensors, actuators, switches and any devices.

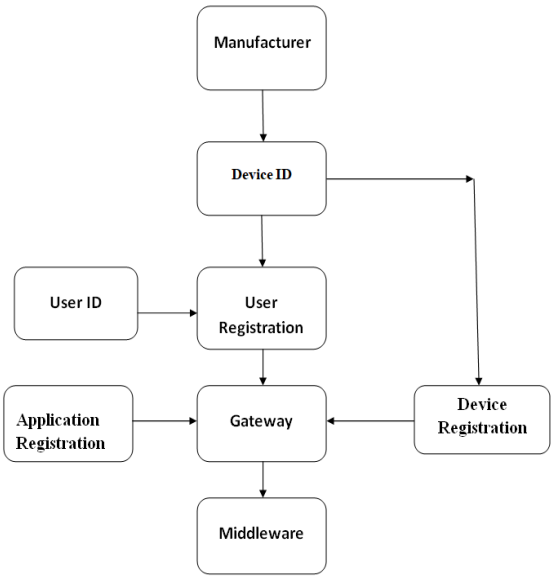

Fig.4. Proposed Work

The proposed model is implemented for the home automation network using ESP8266 and personal data storage. Fig.5. Home automation Network formed for transmission of data from different sensors to application[2]. The MQTT Client and MQTT server registration and steps for authentication and transmission of data is shown in Fig.6.

In Home automation network the data transmission from the node to gateway and internet to the applications will be done using MQTT protocol. The gateway will transmit data in encrypted for middleware formation and follows the HTTP protocol for the transmission of data. The connection from the internet to the application is done using the middleware [2]. Due to this end to end communication confidentiality, integrity and availability of major problem in IoT are rectified. So, even though the attackers try to rectify the data, it cannot be decrypted and the device id, token and user id cannot be traced.

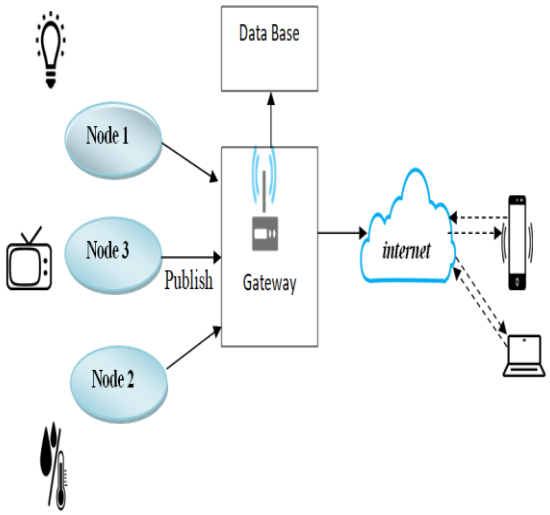

Fig.5. IoT Home Automation Network

After the registration of user, devices and applications to the server and gateway, registration details are stored in the server. As gateway is also registered for identification, the transmission will taken place through the gateway. Gateway will check registration details based on its access token [13].

Before subscription or publication, application and the device have to form the connection with gateway using the token. The gateway checks the registration details in server and then passes on to device id. Device id introspects token and releases the validity for the token. Based on it gateway initializes personal user and forms a communication path.

When Application subscribes for the topic, gateway checks the validity of the user account that is created and connects to that user account [12]. Application doesn't know the user id and it just get connected based on token only. When Publication request is given from the device to gateway, validity for the device located user account and forms a connection with the user account. User account now gives the permission for the transmission to a defined application through the gateway. So gateway transmits data to that application

The publication of data from the device is done automatically based the prescribed interval of time. All time data is stored in data server and only during prescribed intervals of time data is transmitted to subscriber if application is in active state. Once the validity is over or the application unsubscribed or device registration details are varied automatically the connection between the gateway and the Subscriber is removed. Again the application has to register again and using token has to form the new communication path. Received details in the application will remain same. User can view the stored data in server based on the registered user id and password that too from a specified personal network.

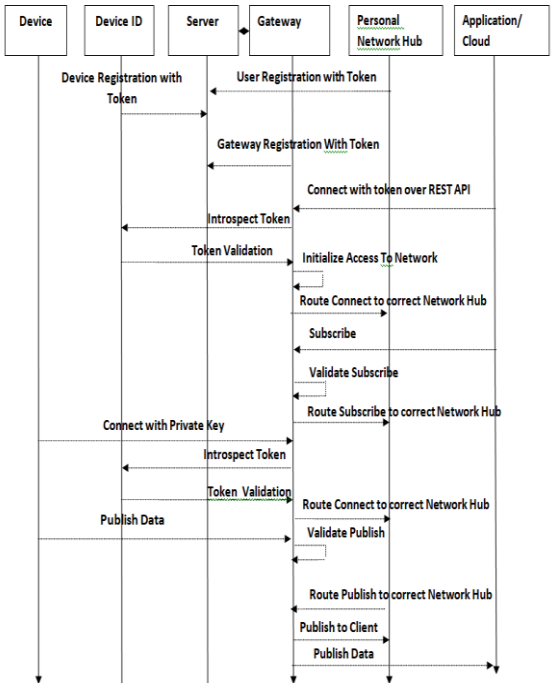

Fig.4. Device Identification, Publication and Subscription to Applications

V. RESULTS

The results give the latency of publication and delivery of messages based on the increase in the client count. The time taken for middleware is less compared to default MQTT and power consumption for the specified home automation network is also very low. transmission is done only during specified intervals of time.

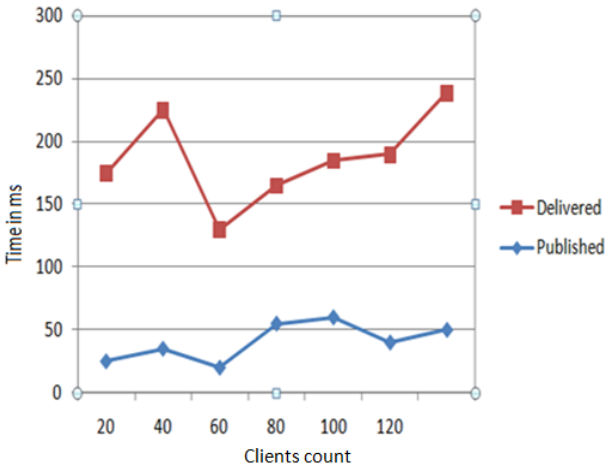

Fig.7. Latency of Publications and Delivery

VI. CONCLUSION

This paper proposed a IoT Middleware Platform and Gateway Model for a Home automation in Iot Applications. REST API is used for end to end secure communication between two devices. By exposing REST API with the middleware, each user, device, application and gateway has to register and after proper authorization using token and identity only data is transmitted device to the application. The devices that are used are of light weight and low cost. Bandwidth utilization and power consumption is also very low.

REFERENCES

1. da Cruz, Mauro AA, et al. "A reference model for internet of things middleware." *IEEE Internet of Things Journal* 5.2 (2018): 871-883.
2. Kodali, Ravi Kishore, and SreeRamya Soratkal. "MQTT based home automation system using ESP8266." *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. IEEE, 2016.
3. Fremantle, Paul, and Philip Scott. "A survey of secure middleware for the Internet of Things." *PeerJ Computer Science* 3 (2017): e114.
4. Luoto, Antti, and Kari Systä. "Fighting network restrictions of request-response pattern with MQTT." *IET Software* 12.5 (2018): 410-417.
5. M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti and M. Jo, "Design of Secure User Authenticated Key Management Protocol for Generic IoT Networks," in *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 269-282, Feb. 2018.
6. Joseph, Tintu, et al. "IoT middleware for smart city:(An integrated and centrally managed IoT middleware for smart city)." *2017 IEEE Region 10 Symposium (TENSYMP)*. IEEE, 2017.
7. Garg, Hittu, and Mayank Dave. "Securing IoT Devices and SecurelyConnecting the Dots Using REST API and Middleware." *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*. IEEE, 2019.
8. Fang, Shifeng, et al. "An integrated system for regional environmental monitoring and management based on internet of things." *IEEE Transactions on Industrial Informatics* 10.2 (2014): 1596-1605.
9. Cheng, Bo, et al. "Lightweight service mashup middleware with REST style architecture for IoT applications." *IEEE Transactions on Network and Service Management* 15.3 (2018): 1063-1075.
10. Tiburski, Ramao Tiago, et al. "The role of lightweight approaches towards the standardization of a security architecture for IoT middleware systems." *IEEE Communications Magazine* 54.12 (2016): 56-62.
11. Ngu, Anne H., et al. "IoT middleware: A survey on issues and enabling technologies." *IEEE Internet of Things Journal* 4.1 (2016): 1-20.
12. Kodali, Ravi Kishore, and Venkata Sundeep Kumar Gorantla. "Weather tracking system using MQTT and SQLite." *2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*. IEEE, 2017.
13. Kodali, Ravi Kishore, and Kopulwar Shishir Mahesh. "Low cost implementation of smart home automation." *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2017.
14. F. D. Hudson, "Enabling Trust and Security: TIPPSS for IoT," in IT Professional, vol. 20, no. 2, pp. 15-18, Mar./Apr. 2018. doi: 10.1109/MITP.2018.021921646.
15. R. T. Tiburski, L. A. Amaral, E. de Matos, D. F. G. de Azevedo and F. Hessel, "The Role of Lightweight Approaches Towards the Standardization of a Security Architecture for IoT Middleware Systems," in *IEEE Communications Magazine*, vol. 54, no. 12, pp. 56-62, December 2016.