Securing Internet of Things Embedded Gateway

Sagar D. Nasre[1], Sunil G.Tamhankar[2]
[1]Dept of Electronics
[2]Assistant Professor, Dept of Electronics
[1, 2] Walchand College of Engineering, Sangli, Maharashtra 416415, India

Abstract- *The Internet of Things (IoT) emerges as a myriad of devices and services that interact to build complex distributed applications. Interoperability and security are imperative for the realization of this vision. Machine-to-machine (M2M) communications standards can be the middleware that glues together the IoT. However, standards are highly complex and require a large amount of interpretation, deployments are currently scarce, and performance evaluations simplistic or speculative. Hence the need for the interconversion of these middleware protocols is essential. Gateway plays a major role in this interconversion. Moreover the data that is being sent through the gateway should be secured so that the outsider cannot read the data that is being sent.* ***Protection of data is prime important, encryption is one of the solution for the same.***

But the encryption algorithm should be lightweight in order to support the energy constrained IoT devices. Thus in this project an attempt is made to design a gateway and implement an appropriate encryption algorithm for securing the data that is being sent through the gateway. We have analyzed the algorithm for test such as avalanche test, and also seen the relation of the cipher text generation on the key to see the strength of the cipher. We have then calculated the execution time to see the response of the algorithm.

Keywords- IoT, Security, Encryption, Wireless Sensor Network WSN, Khazad

I. INTRODUCTION

Over the past few years the world has seen a boom in various internet based technologies, of which one technology had more of a prevailing impact, the technology called the Internet of Things has indeed shown an immense potential for changing the way we live. The primary reason for it to develop was the cheap and easy access to the internet. Initially it was the smartphones that were accessing the internet but now all of the tangible as well as wearable devices are getting connected to the internet. The IOT gained popularity in the year 2010 .IOT is basically controlling a network of physical devices to the internet, but the devices cannot be directly connected to the internet since they communicate on different protocols which operate on short ranges and low power. So to convert the physical world data, Internet Gateway is the mediator which converts the protocols in required form and further carry the data to server/cloud for processing. This can be understood with the help of a simple IOT based Home Automation example wherein all the devices inside the home such as TV, fridge, air conditioners, ovens, doors and windows are all connected to the each other through various protocols like Bluetooth or Wifi (short range protocols) they communicate the data on the internet through a modem, here the modem acts like a gateway sending data signals from the devices to the internet so that user can control it from anywhere sending control signals. Hence gateway plays an important role in an IOT network.

More importantly Gateway can be said as the entry point inside the network, and hence security at the gateway level is an important concern. Any outsider can easily read the system data if the data is not properly secure. The data that is being sent through our homes or hospitals or any other surrounding data should always be private and should be visible only to the authorized user. Thus the gateway should perform two functionalities, i.e. first it should be able to convert the short range protocols to internet based protocols and second the data sent through it should be secured. For securing the data, an encryption algorithm is used. With growing number of devices connected to the IOT platform, it must offer adequate security to their data to encourage the adaptation process.

By and of itself IOT is vulnerable to information theft and may result into crash of the entire communication network. Pwnie Express, an investigative organization focused on cyber threats, few years ago published their third-annual Internet of Evil Things Report, a deep-dive into how IoT devices are becoming increasingly difficult to secure [1]**.** Identity management on the IoT network is different from the workforce or customer identity management. It demands a different and scalable solution with end-to-end encryption to minimize the risk of rogue devices and Man in the Middle attacks. With IoT, security is too important a feature to treat as an afterthought. Currently, while IoT frameworks do have some level of security, it is insufficient to handle sophisticated and highly probable attacks. This means that the risk of hackers and eavesdroppers is huge [2]**.** Hence an appropriate

encryption algorithm should be in place to protect the data and safeguard it from the eavesdropping. Many standard cryptographic algorithms are available in IoT, but there utilization in IoT is questionable as IoT devices demand low power and minimum computational complexity. A trade-off must be done to fulfil the requirement of security with low computational cost.

A. Interoperability issues in IoT platform

Today's world is witnessing a boom of new varieties of consumer electronic devices, sensors, portable devices that are interconnected to each other over the network. Few of which have the ability to connect to a global network (mobile phones, tablets, etc.) while rest are restricted to a more private network such as homes, hospitals and many other.

The emergence of IOT is all about creating a world in which devices, be it wearable devise, security appliances, smart meters or HVACs can communicate among themselves uninterrupted in a transparent way to make our lives simpler. However, interoperability between these devices is a bigger challenge due to IoT fragmentation as each connected equipment's uses a communication protocols that are not interconnected and often not interoperable which is the ground reality and impedes from interconnecting devices on different platforms. Hence interoperability is critical to achieve the much awaited end user experience of IoT [3].

Consider different scenarios for an example:

1) HVAC system that understands and adjusts the temperature preferences depending on your mood.
2) Refrigerators which sends a message to restock or recommends changing a derelict part.
3) Security systems that can identify humans and employ machine learning to figure out any abnormal activity.

B. Security Issues in IoT

In order to make IoT network globally recognized it is necessary that it provides security and privacy of the user's data. IoT devices remain unsupervised over a long duration of time, so there is always a fair chance of physical attack on its components. Also due to wireless communication medium, the eavesdropping is extremely simple. IoT devices are constrained in the terms of energy and computational power; hence it is not possible to implement huge and complex security algorithms on them. The use of computationally expensive security algorithms will only hinder the performance and will consume a lot of energy is doing so. But data integrity and credibility are the areas of concern. Hence a lightweight algorithm is required when constrained IoT devices are concerned.

C. Motivation and Organisation of Paper

Statistics have suggested that there will be billions of devices that are going to be connected to the internet by 2020, and the middleware devices like the Gateway is going to play a major role in it. Also these devises are vulnerable to various kinds of attack; these attacks can be initiated by sensing the communication between the nodes which is known as Man-in-Middle attack. There is no reliable solution available to handle such attacks. However, a lightweight encryption technique could minimize the amount of damage that is done to the data credibility and integrity. Even though the conventional cryptographic algorithm provides a good level of security, they are not well suited for the constrained IoT devices. Hence a trade-off must be done to fulfil the requirements of security with low computational costs.

In this paper, we have proposed an implementation of a gateway which sends the sensor data to the cloud using simple MQTT protocol and also a lightweight encryption algorithm is used to secure the data from security threats as mentioned above. The rest of the paper is organized as follows, section II talks about how the general model is implemented. Section III and IV talks about the actual implementation as to how the setup is developed finally the results and the discussion on the results is done in section V, based on the results conclusions are obtained in section VI and finally the future scope in section VII is discussed.

II. MODEL

The system mainly is divided into two parts, in first part we have developed a sample scenario where sensor data (in this case light intensity of the surrounding) is being sent to the cloud server through a gateway (Intel Galileo Board) using MQTT protocol. MQTT stands for Message Queue Telemetry Transport and it is a protocol for transporting messages between two points. The reason for using MQTT in IoT applications is because of its super lightweight architecture, which is ideal for scenarios where bandwidth is limited. The flowchart for the said scenario is shown below in Fig.1.

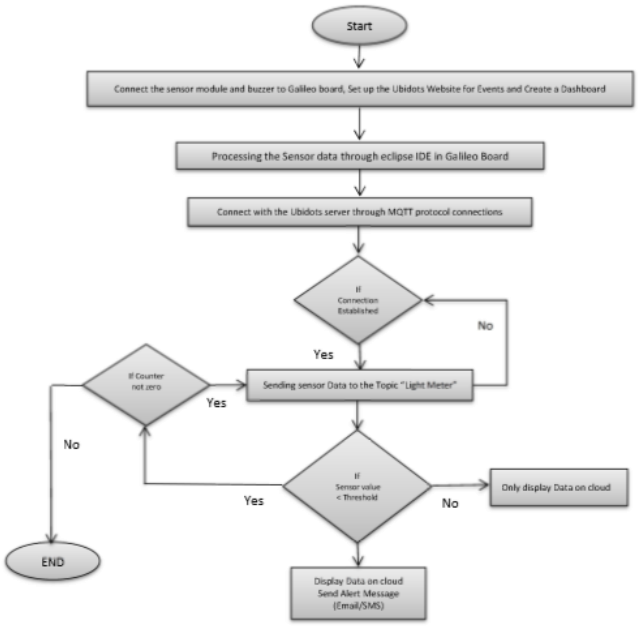

Figure 1. MQTT protocol communication

In the second part we are using a lightweight encryption algorithm for the MQTT data packets that are passing on the internet through the Gateway.

III. IMPLEMENTATION

The below subsections explains the approaches used to implement the system

A. Implementation of Gateway

Gateway being the exit point in the network it needs to send the data to the server ("things.ubidots.com"), which is an open-source server-side platform that allows you to monitor and control IoT devices. It is free for both personal and commercial usage and you can deploy it anywhere. Steps for configuring the Intel Galileo board as a gateway are as follows:

1. Make sure that the target is completely powered off before performing the steps described below
2. Also make sure that the antennas are connected on the gateway platform before performing the steps mentioned below
3. Boost the host machine a.k.a. computer from the pen drive
4. Connect the FTDI to USB converter cable to one of the USB ports on your host machine
5. Open the terminal on the host machine and connect the gateway to the Wi-Fi

Sequence of commands is used to connect gateway to Wi-Fi. Check for IP connectivity with required commands. Procedure for setting up of Eclipse IDE which will be used for writing the code

a) Launching the Eclipse IDE
b) Opening a project and building
c) Configuring the target in the Eclipse IDE
d) Loading the project on the target and running
e) Building the project and debugging the same
f) Setting breakpoints and single stepping through the code

The main function in the code does the following tasks

a) Confirms the target is an Intel Galileo or Edison platform
b) Create a client for connecting to the Ubidots server

```
rc = MQTTClient_create(&client, const_cast<char *>(host), const_cast<char *>(clientID),
MQTTCLIENT_PERSISTENCE_NONE, NULL);
if (rc != MQTTCLIENT_SUCCESS)
{
std::cerr << "Failed to create MQTT client, exiting : " << rc << std::endl;
exit(rc);
}
```

c) Setup call backs before connecting the client to the server

```
MQTTClient_setCallbacks(client, NULL, &connection_lost, &message_arrived, &delivery_complete);
MQTTClient_connectOptions data = MQTTClient_connectOptions_initializer; data.username = const_cast<char *>(ubidots_username);
data.password = NULL;
```

d) Connect the client to the server

```
rc = MQTTClient_connect(client, &data);
if (rc != MQTTCLIENT_SUCCESS)
{
        std::cerr << "Failed to connect MQTT client, exiting :" << rc <<std::endl;
        exit(rc);
}
```

e) Subscribe to the variable

```
rc = MQTTClient_subscribe(client, const_cast<char *>(TOPIC), MQTT_DEFAULT_QOS);
if (rc != MQTTCLIENT_SUCCESS) {
```

```
std::cerr << "Failed to subscribe, exiting :" << rc <<std::endl;
        exit(rc);
}
```

f) Enter loop to read light sensor

```
rc = MQTTClient_subscribe(client, const_cast<char *>(TOPIC), MQTT_DEFAULT_QOS);
if (rc != MQTTCLIENT_SUCCESS) {
        std::cerr << "Failed to subscribe, exiting :" << rc <<std::endl;
        exit(rc);
}
```

g) Publish to the cloud and Stop after sending 60 values

```
int rc = MQTTClient_publish(client, const_cast<char *>(MESSAGE_TOPIC), message_size, const_cast<char *>(payload), MQTT_DEFAULT_QOS, retained, &dt);
                if (rc == MQTTCLIENT_SUCCESS)
{
printf("Waiting for message with token %d to be published...\n", dt);
rc = MQTTClient_waitForCompletion(client, dt, 1000);
if (rc == MQTTCLIENT_SUCCESS)
{
printf("Message with token %d published\n", dt);
        }
                else
        {
std::cerr << "Failed to publish message with token " << dt << std::endl;
                        }
        }
                else
        {
std::cerr << "Failed to publish message with token " << dt << std::endl;
        }
        sleep(2);
        }
        printf("Stopping\n");
// deleting all the variable that are created and exiting from the connection
        int timeout = 100;
MQTTClient_disconnect(client, timeout);
MQTTClient_destroy(&client);
        delete sound;
        delete screen;
delete light;
```

B. Data Packets visualization using WIRESHARK software

Wireshark is one of the best network data visualization tool. Two python scripts i.e. Publisher.py and Subscriber.py are created. In publisher.py script we have used Paho MQTT broker which is located at “iot.eclipse.org”, the default port used is 1883 and the keep alive interval is 10 seconds so that we will easily be able to capture the ping messages. Here the topic we will publish and subscribe is “SampleTopic” and the message that will be published is “Hello MQTT”. Before running both the scripts we will start Wireshark and start capturing Data packets.

Steps to capture Data packets:

1. Start an instance of Wireshark
2. Run Subscriber.py script
3. Run Publisher.py script
4. Stop the instance of Wireshark.

Wireshark has captured the MQTT data packets. The captured MQTT data packets are presented in the Fig.2.

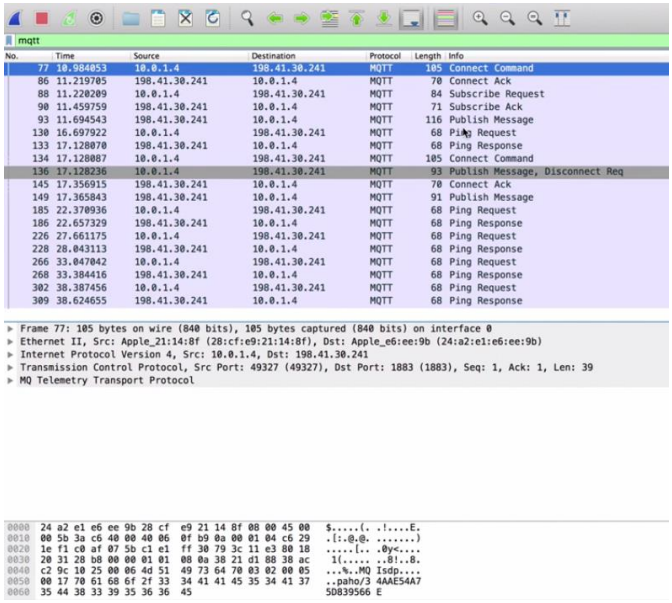

Figure 2 MQTT packet capture

C. Implementation of security layer

The security is implemented at the gateway which is the exit point of the network. The string of data that is to be sent is feed to the encryption algorithm for generation of cipher text. The lightweight algorithm that we are using is Secure Internet of Things (SIT) mentioned in paper of Muhammad Usman et.al. [4]. In the paper they have implemented the algorithm on images and shown the results in

the form of encrypted and decrypted images. The flowchart for the said algorithm is shown in fig.3.

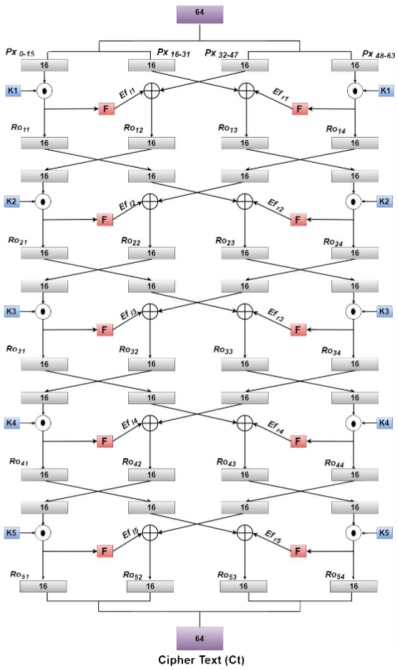

Figure 3

The algorithm creates confusion and diffusion of logical operations, left shifting, swapping and substitution.

IV. EXPERIMENTAL SETUP

A. Intel Galileo as a Gateway

Fig.4 shows the Intel Galileo board sending data to the Ubidots server. A light sensor, LCD module and a buzzer is connected to the board. Antennas connected to the board gives wireless connection capabilities to the board.

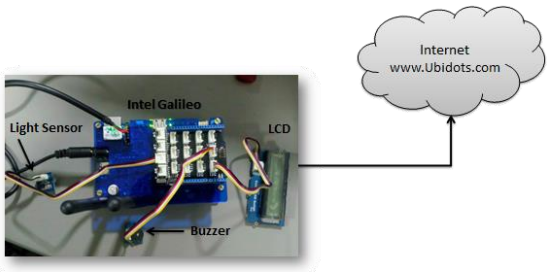

Figure 4 Intel Galileo as a Gateway

The figures below show the light intensity in a graphical form on Ubidots server. Fig.5 shows the light intensity values in Gauge indicator and the Fig 6. Shows the value in a graph. There are various options available to indicate the sensor values on the Ubidots Server Dashboard.

Figure 5 Gauge meter

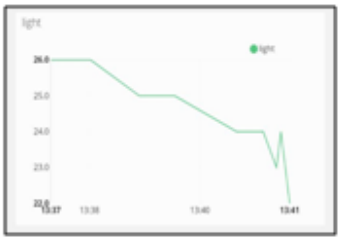

Figure 6 Graph of light values

The figure below displays the events that are created on Ubidots server for continuously monitoring the light intensity as well as sounding an alert messages and firing up a buzzer. The condition is set in such a manner that if the light intensity values that are received from the gateway are set below 10 then the user gets a mail and sounds a buzzer.

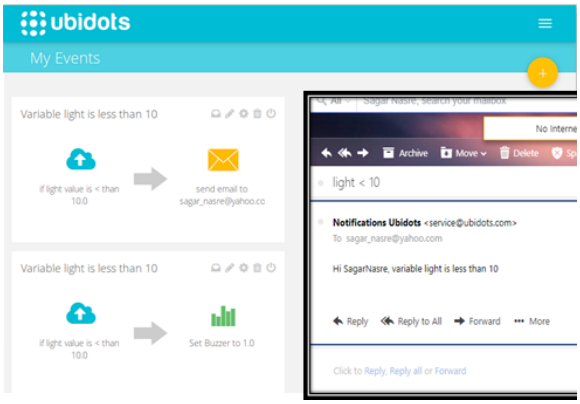

Figure 7 Screenshot of email

B. Wireshark Packet capture

The figure below shows the Wireshark instance where the MQTT packets are captured and displayed. From the captured instance it is clear that MQTT messages sent can be easily viewed through the Wireshark software.

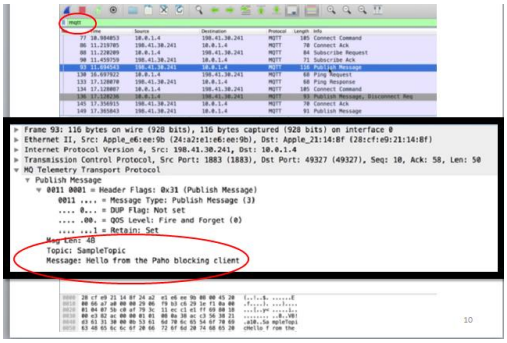

Figure 8 Message Capture in Wireshark

C. Results from the encryption algorithm

A sample word "Hello MQTT" was used for the encryption, the word can be considered as a MQTT publish message which the publisher script is sending to the subscriber script. Any sample string can be taken as an input and can be feed to the SIT: Lightweight Encryption Algorithm. Below Fig.9 shows the histogram of input variable "Hello MQTT" and resultant histogram of the variable that is encrypted and again extracted variable "Hello MQTT" from the decryption algorithm.

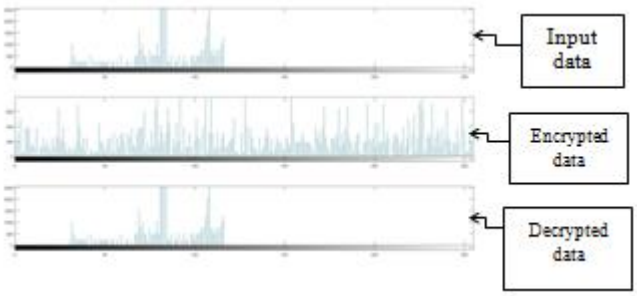

Figure 9 Histogram of results

We have also done the correlation comparison in Fig.10 which illustrates the contrast between the original and encrypted data.

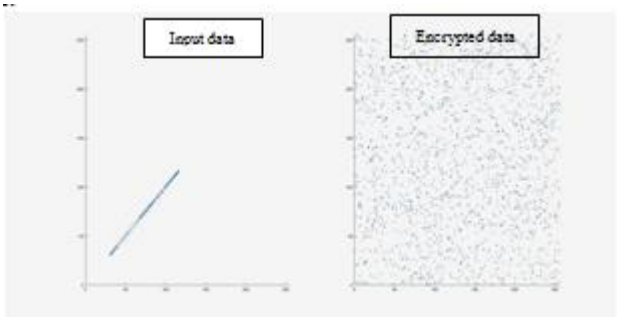

Figure 10 Scatter plots

We have performed the Avalanche Test for measuring the strength of the encryption algorithm. For this we had changed a single variable in our key, and we have checked the resultant effect on the cipher text. The screenshot of the same is shown in the Fig.11, it can be seen from the calculations that the percentage change in the cipher text for a single character change in the key is about 59.27 %.

The execution time has also been calculated using the tic and toc function in MATLAB which gives the exact time for encrypting and decrypting the information and comes out to be around 0.48 milliseconds on an average.

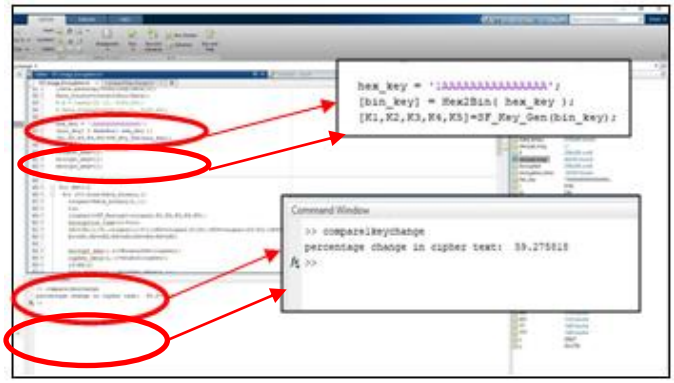

Figure 11. Results of key change

V. CONCLUSION

Gateway can be configured on the Intel Galileo Board and using MQTT broker client communication, the sensor data can be successfully sent to the Ubidots cloud server, also a varied range of graphical indicators are present on the Dashboard of the Ubidots server so that a graphical and continuous monitoring of data is possible. MQTT is a very lightweight messaging protocol which communicates with minimum number of commands and has a very fast response time, and thus MQTT is well suited protocol for IoT and M2M communications and its applications are endless.

MQTT is good and lightweight messaging protocol for IoT, but the data is not secure for which modified SIT encryption algorithm is used. Modified SIT encryption algorithm has shown promising results and thus is one of the suitable solutions.

Modified SIT adds randomness to the data input and that there is a uniform distribution of intensities after the encryption, which is an indication of desired security as we cannot extract the intensities of the data inputs from the encrypted image. The decryption algorithm successfully extracts the original data points. The modified SIT algorithm has a good response time and satisfies the Avalanche test showing good strength.

From the above discussion it is clear that Internet of Things is going to have a huge impact on the way we send data among each other and with the Things around us, and security of this data is of utmost important. For this purpose a lightweight security algorithm is proposed named as modified SIT which can be implemented on the IoT Gateway.

VI. FUTURE SCOPE

For future research, the implementation of the algorithm on hardware and software in various computation

and network environment is under consideration. Moreover, the algorithm can be optimized in order to enhance the performance according to different hardware platforms. Hardware like FPGA performs the parallel execution of the code, the implementation of the proposed algorithm on an FPGA is expected to provide high throughput.

The scalability of algorithm can be explored for better security and performance by changing the number of rounds or the architecture to support different key length.

REFERENCES

[1] https://channels.theinnovationenterprise.com/articles/4-statistics-that-reveal-major-problems-with-IoT-security.
[2] https://tech.economictimes.indiatimes.com/news/technology/IoT-needs-to-adhere-to-robust-policies-to-avoid-security-risks/64283644
[3] https://connect.aricent.com/2017/04/interoperability-challenges-in-IoT/
[4] Muhammad Usman, Irfan Ahmed, M. Imran Aslam, Shujaat Khan and Usman Ali Shah, "SIT: A Lightweight Encryption Algorithm for Secure Internet of Things", IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 1, 2017, pp. 402-411

At least two or more references from good journal / conf.