IOT BASED ACCESS MECHANISM USING INTER PLANETARY FILE SYSTEM

A PROJECT REPORT

Submitted by

CB.EN.U4CSE15422	**K. PANDU RANGA REDDY**
CB.EN.U4CSE15358	**V. SAI ABHINAV REDDY**
CB.EN.U4CSE15522	**D. KRISHNA CHAITANYA VARMA**
CB.EN.U4CSE15312	**D. VAMSI KRISHNA REDDY**

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF ENGINEERING, COIMBATORE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE 641 112

April 2019

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, COIMBATORE, 641112

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"IOT BASED ACCESS MECHANISM USING INTER PLANETARY FILE SYSTEM"** submitted by K. Pandu Ranga Reddy (CB.EN.U4CSE15422), V. Sai Abhinav Reddy (CB.EN.U4CSE15358), D. Krishna Chaitanya Varma (CB.EN.U4CSE15522), D. Vamsi Krishna Reddy (CB.EN.U4CSE15312) in partial fulfillment of the requirements for the award of the Degree **Bachelo**r **of Technology** in **Computer Science and Engineering** is a bonafide record of the work carried out under our guidance and supervision at Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore.

INTERNAL SUPERVISOR

Dr. S Thangavelu
Assistant Professor
Dept. of Computer Science and Engg .

CHAIRPERSON

Dr. Latha Parameswaran
Dept. of Computer Science and Engg.

This project report was evaluated by us on :……………………………..

INTERNAL EXAMINER

EXTERNAL EXAMINER

TABLE OF CONTENTS

ACKNOWLEDGEMENT

We express our gratitude to our beloved **Satguru Sri Mata Amritanandamayi Devi** for providing a bright academic climate at this university, which has made this entire task appreciable. This acknowledgement is intended to be a thanks giving measure to all those people involved directly or indirectly with our project. We would like to thank our Vice Chancellor **Dr. Venkat Rangan. P** and **Dr. Sasangan Ramanathan** Dean Engineering of Amrita Vishwa Vidyapeetham for providing us the necessary infrastructure required for completion of the project.

We express our thanks to **Dr. Latha Parameswaran**, Chairperson of Department of Computer Science Engineering, Prof. Prashant R Nair and Dr. P Bagavathy Sivakumar, Vice Chairpersons of the Department of Computer Science and Engineering for their valuable help and support during our study. We express our gratitude to our guides, **Dr. S Thangavelu,** for his/her guidance, support and supervision.

We feel extremely grateful to **Dr. D Venkataraman, Ms. K Nalina Devi, Mr. C Arun Kumar, Ms. R Manjusha** for their feedback and encouragement which helped us to complete the project. We also thank the staff of the Department of Computer Science Engineering for their support.

We would like to extend our sincere thanks to our family and friends for helping and motivating us during the course of the project. Finally, we would like to thank all those who have helped, guided and encouraged us directly or indirectly during the project work. Last but not the least, we thank **God** for His blessings which made our project a success.

ABSTRACT

In the blooming era of Internet of Things, trust has been accepted as a vital factor for provisioning secure, reliable, seamless communications and services. However, these IoT devices use various protocols to communicate in which most of them follow a centralized architecture for data transfer. Some of these protocols are MQTT which has broker for decoupling and transferring data between publisher and subscriber. These centralized servers can be third party service providers which rises the issue of trust. To address this a decentralized and distributed network architecture called IPFS is used instead of third-party service providers. IPFS enables publish – subscribe model similar to MQTT but by removing the broker and making a peer-to-peer interaction. Peer-to-peer architecture has its own challenges like security which can be addressed by encrypting the files or data that has to be sent over the IPFS network using PGP (Pretty Good Protocol).

LIST OF FIGURES

LIST OF ABBREVIATIONS

IOT	Internet of Things
IPFS	Inter Planetary File System
WSN	Wireless Sensor Networks
MQTT	Message Queue Telemetry Transport
PGP	Pretty Good Privacy
GPG	GNU Privacy Guard
DHT	Distributed Hash Table
DAG	Directed Acyclic Graph

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

The Internet of Things is a global industry movement that brings together people, process, data and things to make networked connection more relevant and valuable than ever before. There are millions of IoT devices connected now and are estimated to grow rapidly to billions in next few years. Considering all these facts security plays a major role in communication between devices. There are many protocols that IoT devices use in which most of them are centralized which have intermediate node (broker) for communication. This broker has to know the

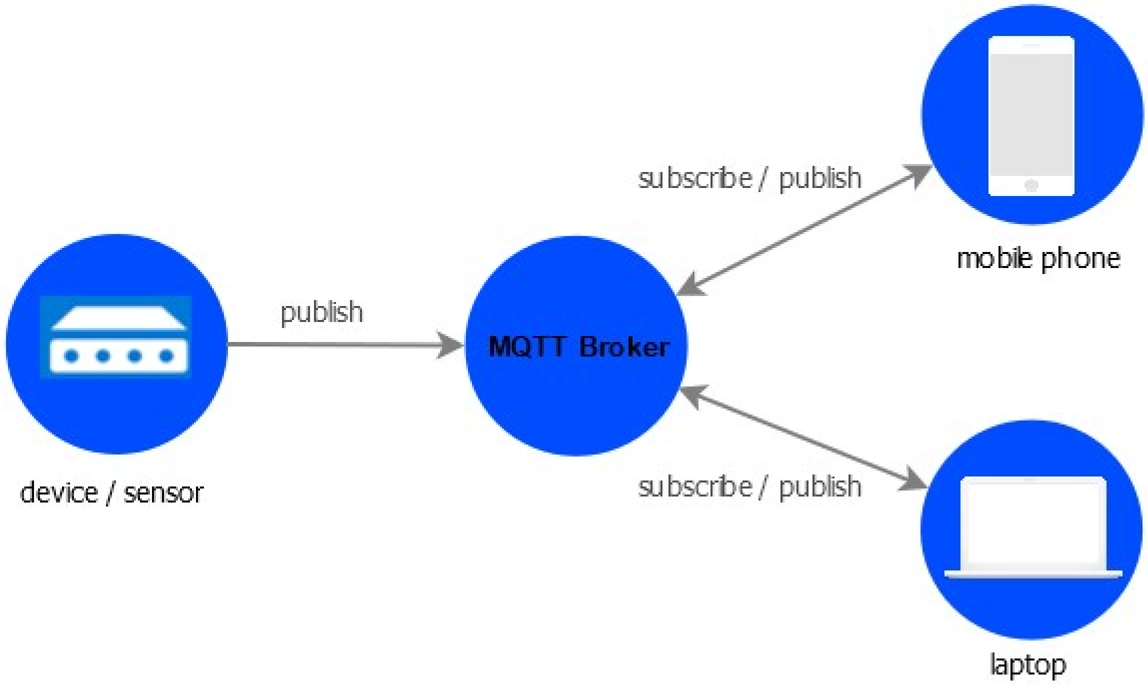


Fig. 1.1 – MQTT publish subscribe using broker as intermediate node

Considering all the those specified above, a decentralized and distributed architectural based protocol like IPFS can be used to address the issue of broker. IPFS is a protocol and network designed to create a peer-to-peer method of storing and sharing hypermedia in a distributed file system. It provides high performance and clustered persistence. While delivering large amount of data to users, a peer-to-peer approach could save millions of bandwidths. High latency networks are a real barrier of entry to developing world. IPFS provides resilient access to data, independent of low latency or connectivity to the backbone. It has a publish-subscribe model which focuses on reliability, delivery guarantee and data persistency. Publish-Subscribe, called ‘pubsub’ for short, is a pattern often used to handle events in large-scale networks. ‘Publishers’ send messages classified by topic or content and ‘Subscribers’ receive only the messages they are interested in, all without direct connections between publishers and subscribers. This approach offers much greater network scalability and flexibility. Using this pubsub can ignore broker and provide the same data transfer without or rather more efficiently.

1.2 PROBLEM STATEMENT

Establishing a secure connection from an end application to an IoT device without depending on third party services like cloud, centralized services, etc. where the privacy is not guaranteed.

1.3 SPECIFIC OBJECTIVE

There are lots of different solutions exist in the field. However, most of them either rely on a centralized or hierarchic network to have a reliable system, with stronger delivery and persistence guarantees, or end up sacrificing these same properties in order to have a decentralized system with the potential to scale to a much larger network.

The solution we propose is a pub-sub module to IPFS with a string focus on reliability, delivery guarantees and data persistence, while maintaining the ability to scale to a vast number of users, using the network infrastructure we have in place today.

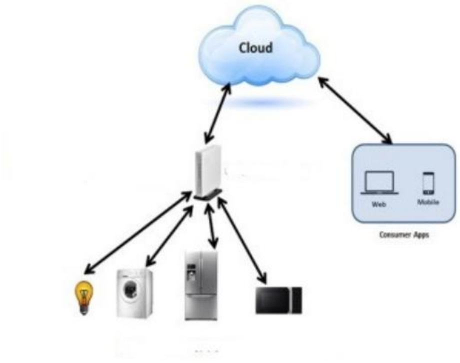


Fig. 1.2 – Using third party cloud

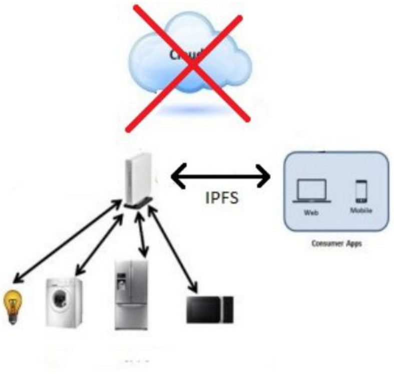


Fig. 1.3 – Using IPFS

1.4 LIMITATIONS

Though the proposed models provide solution for establishing a secured connection from an end application to an IoT device without depending on third party service providers by using Peer-to-Peer architecture, it has some limitations.

- It requires more network resources.
- While finding path for the first time between end applications, it requires more time as it has to find the shortest path possible.

CHAPTER 2

LITERATURE SURVEY

Many researches and authors have long been using MQTT for IoT and most of them stressed on privacy and security. In [7], ensuring privacy for IoT was given higher priority by applying ABE, grounded on nonspecific Pub-Sub architecture. In this technique a payload is used to encrypt by using Advance Encryption Standard (AES) algorithm which comes under Symmetric key cryptography is used to encrypt the payload and to make sure that the payload size and cipher text size as same, AES key is encrypted with the help of ABE. In [8] the authors argue that, both these encryption techniques i.e. AES and ABE are used to accomplish encryption on limited bits of data which are generated by the IoT will be a computational overhead for IOT devices. Hence in [8] the authors aimed at the optimization of ABE's complex arithmetic operations using suitable cryptography parameters (MQTT-S) instead of performing double encryptions. In [9] the authors proposed a middleware based on Pub-Sub architecture in which CP-ABE and Predicate based encryption is enabled to protect the privacy of subscriber's interest and the published content's confidentiality. In [10] the authors present the strategy and execution of IoT based Home Automation, a steadfast WSN technology will be interconnected through MQTT (Telemetry Transport) protocol to establish the communication among diverse devices via "Ngrok" which is a third-party cloud access provider where the author completely trusts their services, which might lead to a point of failure and does not unravel the speed of light problem.

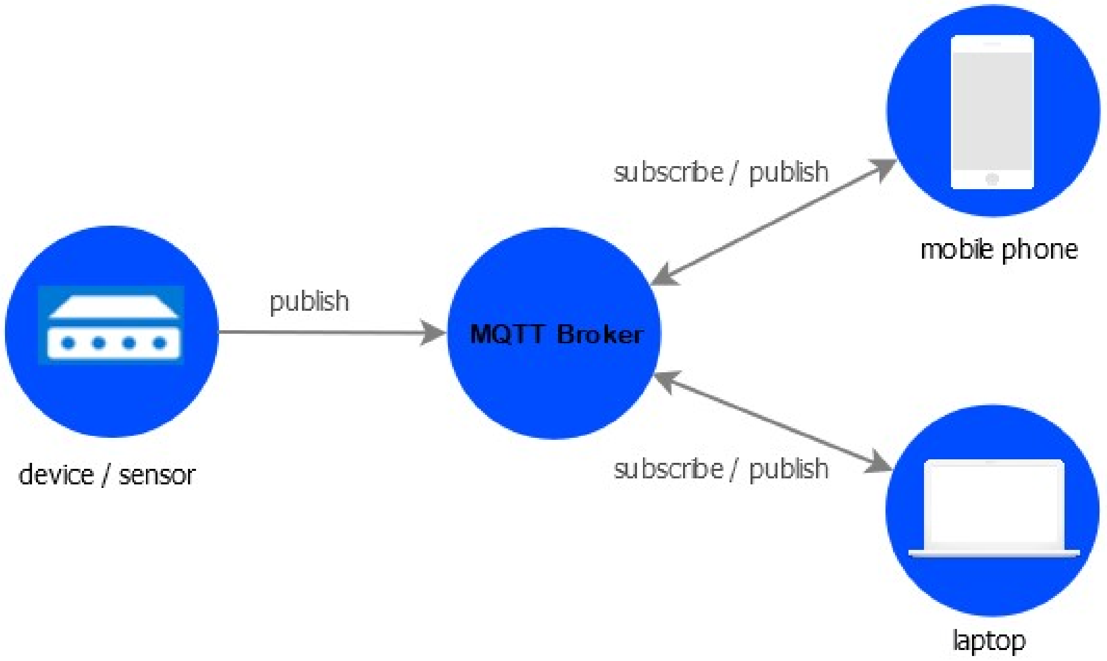


Fig. 2.1 – MQTT publish subscribe model

The challenging aspect in IoT is to introduce privacy for users' data. So, it is necessary to develop a decentralized IOT, which should be designed to have a built-in privacy [11]. With the decentralized IoT data management users will have a choice to share and sell the sensor data to the third-party entities without any need of intermediaries. According to the authors [11] the objective therefore, is to ensure that the user data is not delivered to the centralized entities by providing a decentralized data access model for IoT. To realize this goal the Blockchain techniques and peer-to-peer communication will play a vital role [12], [11]. The Inter Planetary File System, "IPFS", [13] is a cutting-edge peer-to-peer distributed file system that seeks to connect all computing devices with the similar system of files i.e., By trading objects with each other. Unlike all other Telemetry Transport protocol IPFS also has a publish-subscribe feature which does not require a broker for data transmission. Which makes it more resilient to several network based criticization. IPFS has no single point of failure, and nodes do not need to trust each other which it a perfect alternative for MQTT protocol in our research work.

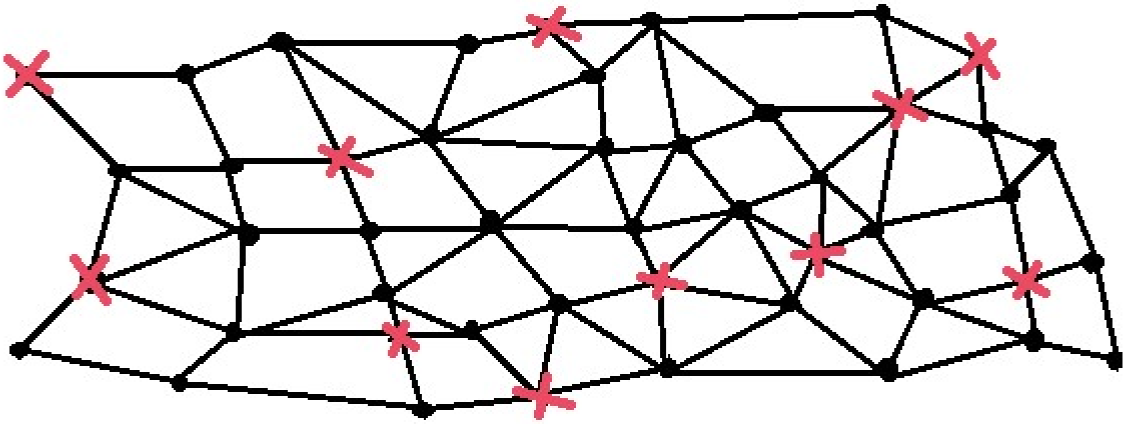


Fig. 2.2 – Distributed Network

As previously discussed, centralized IoT devices will have scalability issues in regard to data management and access control which force the users to give their entire data to the third-party brokers or intermediaries for managing their data and thus loosing the data privacy [14]. This problem has lead to the research on working with the blockchain technology with the help of peer-to-peer data storage mechanisms [11]. Bitcoin's success is the main motivation behind the research on using the blockchain and peer-to-peer technology. In [15] the authors have stressed on the capability of blockchains in maintaining the data exchanges through an immutable log as well as performing the access control. Access policies around the public key infrastructure of blockchain networks have been created from where the access control element comes from [16]. IBM Adept [17] [11] is a combined effort from IBM and Samsung whose aim is to connect the blockchain and to develop a decentralized platform for IoT. For peer-to-peer communication Adept uses Tele Hash, and for peer-to-peer file sharing, an Ethereum blockchain development platform on top of BitTorrent is used. According to [11] "the issues IBM Adept faces in implementing a blockchain based solution for decentralizing IoT are the poor scalability of blockchains and the inherent latency in blockchain consensus". The authors in [15] suggest dividing the IoT blockchain network into smaller sub-networks, since a single blockchain cannot take the load [18]. The Author has explored about various aspects of the file system i.e., which implies security, transparency and data privacy of instigating along with The Internet of Things for a better data and healthy algorithm opacity which lacks the internal protocol security i.e., Encryption. This problem can be solved by open source tool OpenGPG. Thus, captivating all these characteristics into contemplation, A typical order of stages for out problem statement is Initialize the IPFS Daemon–ipfsinit(), Setting the Topic for publish–subscribe, Add Listeners, Encrypt Channel data by Asymmetric Key Encryption, Run scenarios using various cases (Related to accessibility). Impact of Spoofing on IoT devices are one of the major problems and can be secured by cryptographic algorithms [20].

Thus, captivating all these characteristics into contemplation, A typical order of stages for our problem statement is:

1. Initialize the IPFS Daemon – ipfs init().
2. Setting the Topic for publish – subscribe.
3. Encrypt Channel data by Asymmetric Key Encryption.
4. Run scenarios using various cases (Related to accessibility).

CHAPTER 3

SYSTEM SPECIFICATIONS

The proposed system requires the following specifications

Client-Node:

Linux Operating System (Debian)

Go version 1.7 or High

IPFS version 4.17

NetBeans IDE version 9

IoT-Node:

Raspberry Pi 2

Raspbian Operating System version 9

Go version 1.7 or High

IPFS version 4.17

NodeJs v10.8

wiringPi version 2 or High

Tower Pro SG90 Servo Mini/Micro Servo Motor

CHAPTER 4

ARCHITECTURE

4.1 BASIC ARCHITECTURE (IPFS, PUB-SUB AND PGP)

- **IPFS**

The Inter Planetary File System (IPFS) is a distributed file storage protocol that allows computers all over the world to store and serve files as part of a giant, peer-to-peer network. It is also called as Distributed Web. Every single computer that is running IPFS acts as both a client and a server. In other words, each computer running the IPFS software can serve content to any other computer in the network, as well as request content from anyone int the network. Every file added to IPFS is given a unique address derived form a hash of the file's content. IPFS addresses are multi-hashes, which combine information about the hashing algorithm used as well as hash output into a single string.

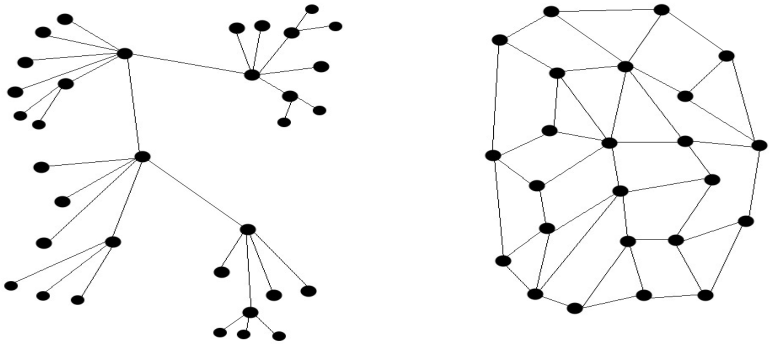


Fig. 4.1 – Decentralized and Distributed Networks

IPFS uses Merkle data format for hashing. Every Merkle tree is a directed acyclic graph (DAG) because each node is accessed via its name. Each branch of

Merkle is the hash of its local contents, naming children by their hash instead of their full contents. So, after creation there is no way to edit a node. This prevents cycles, since one cannot link the first created node to the last node to create the last reference.

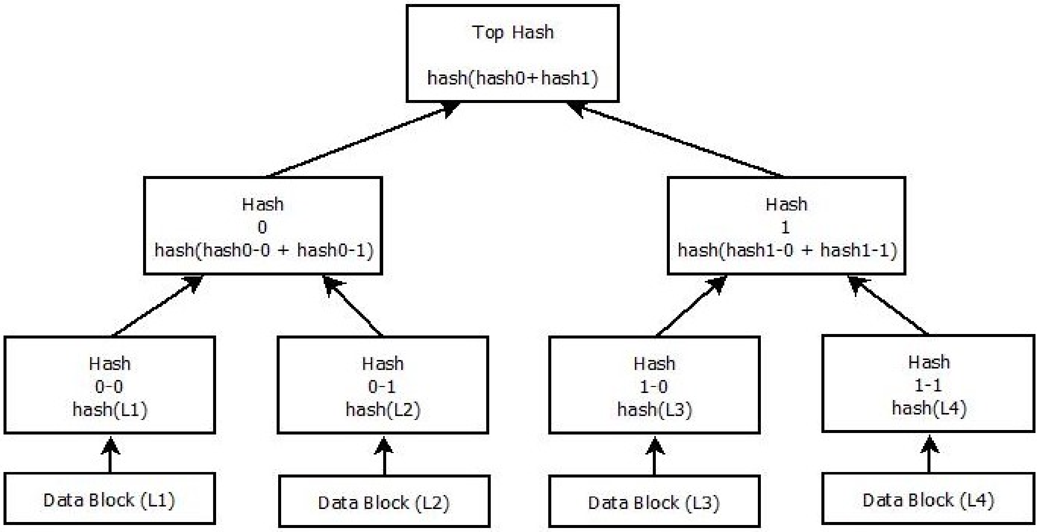
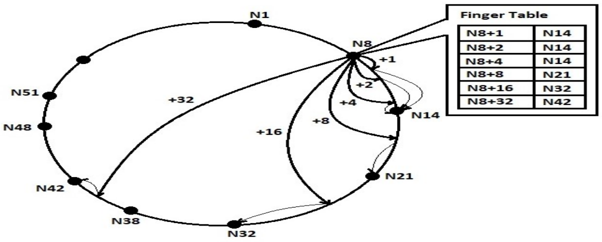


Fig. 4.2 – Merkle DAG Hashing Method

IPFS relies on a distributed hash table (DHT), i.e., a mapping from hash to some people who may have the content addressed by that hash. The hash table is distributed because no single node in the network holds the whole thing. Instead, each node stores a subset of the hash table, as well as information about which nodes are storing other relevant sections.



Finger Table

N8+1	N14
N8+2	N14
N8+4	N14
N8+8	N21
N8+16	N32
N8+32	N42

Fig. 4.3 – Distributed Hash Table

- **PUB-SUB MODEL**

Publish-Subscribe, called ‘pubsub’ for short, is a pattern often used to handle events in large-scale networks. ‘Publishers’ send messages classified by

topic or content and ‘Subscribers’ receive only the messages from the ‘Topics’ they are interested in, all without direct connections between publisher and subscribers. This approach offers much greater network scalability and flexibility.

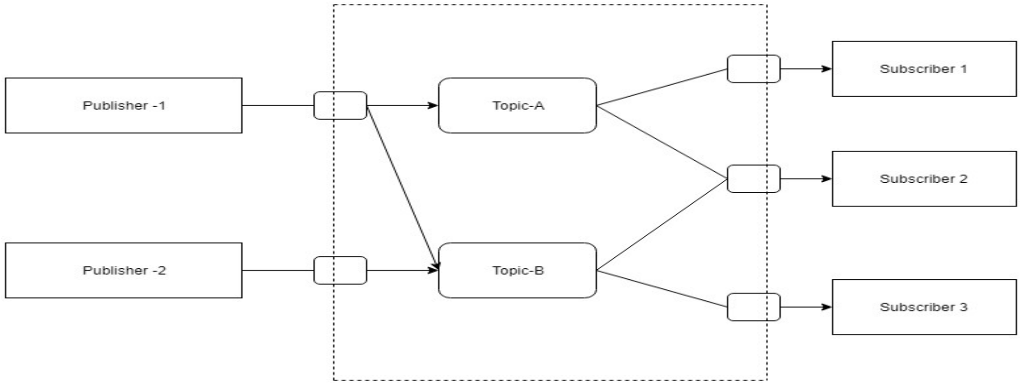


Fig. 4.4 – Publish Subscribe Via Topic Model

- **PGP & GPG**

PGP (Pretty Good Privacy) uses a combination of encryption methodologies such as hashing, data compression, symmetric-key cryptography and public-key cryptography to keep data secure. Each public key is bound to a username or an email address. This process can be used to encrypt text files, emails, data files, directories and disk partitions. OpenPGP is a standard of PGP that is open-source for public use. The GPG software is an independent implementation of the OpenPGP standards, so one can use it to exchange encrypted messages with people using other OpenPGP implementations.

PGP can be used to send messages confidentially by combining symmetric-key encryption and public-key encryption. The message is encrypted using a symmetric encryption algorithm, which requires a symmetric key. Each symmetric key is used only once and is also called a session key. The message and its session key are sent to the receiver. The session key must be sent to the receiver so they know how to decrypt the message, but to protect it during transmission it is encrypted with the receiver's public key. Only the private key belonging to the receiver can decrypt the session key.

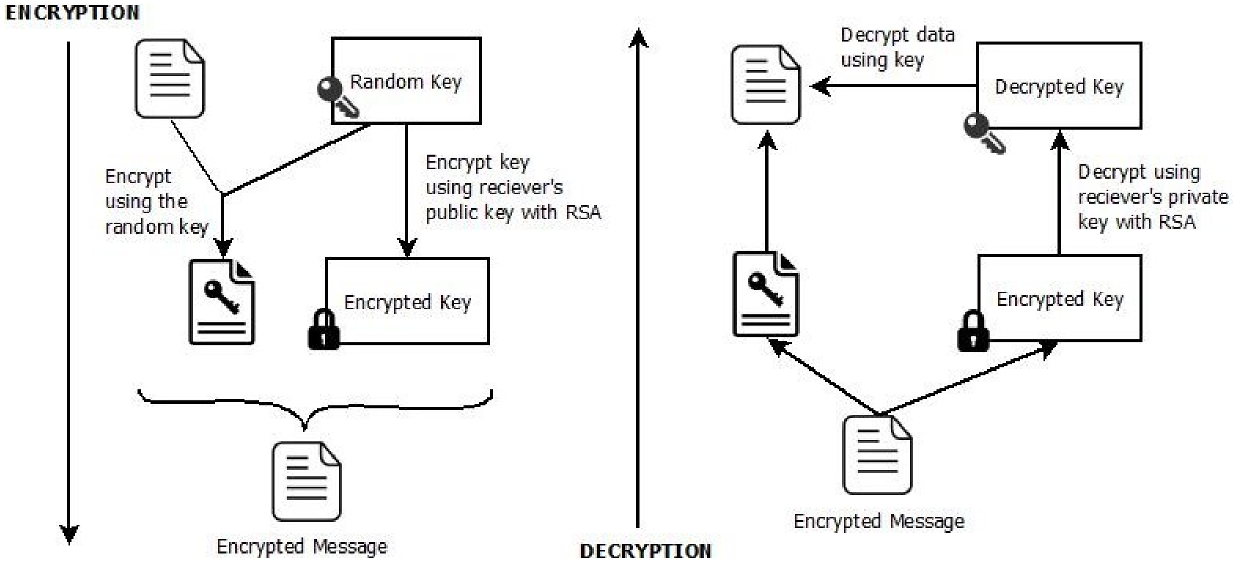


Fig. 4.5 - PGP Encryption and Decryption

It also supports message authentication and integrity checking. Because the content is encrypted, any changes in the message will result in failure of the decryption with the appropriate key. The sender uses PGP to create a digital signature for the message with either the RSA or DSA algorithms. To do so, PGP computes a hash from the plaintext and then creates the digital signature from that hash using the sender's private key.

4.2 PROJECT ARCHITECTURE

STAGE 1: (Publish - Subscribe)

Enabling Publish Subscribe between the end user (PC) and IoT device (Raspberry Pi 2).

When a message is published from the PC, it will be received by both the PC and the Pi as PC is self-subscribed to the topic and Pi is also subscribed to that topic.

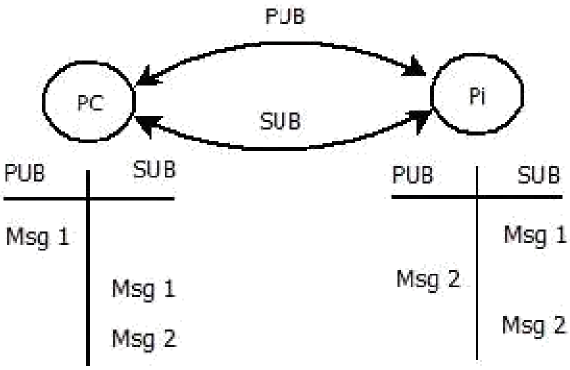


Fig. 4.6 - Pub-Sub between PC and Pi

STAGE 2: (Transfer Files over IPFS)

Adding files to IPFS network and getting the hash for the uploaded file and publishing it to the IPFS network through pub-sub model.

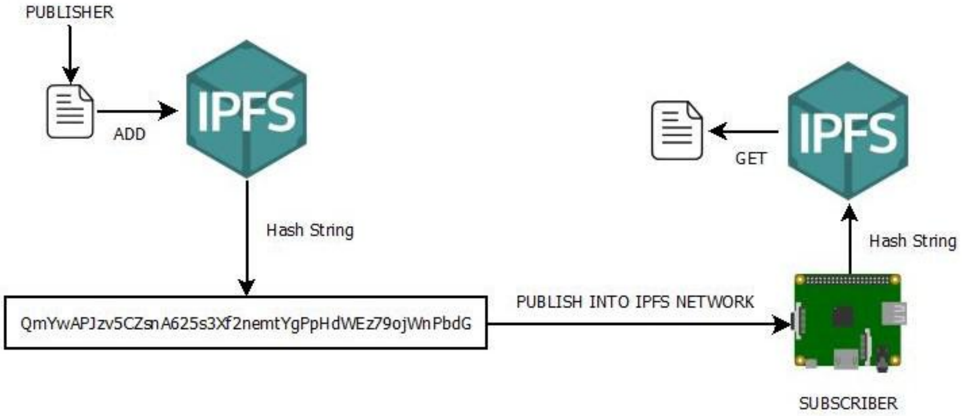

PUBLISHER
ADD
IPFS
Hash String
QmYwAPJzv5CZsnA625s3Xf2nemtYgPpHdWEz79ojWnPbdG
PUBLISH INTO IPFS NETWORK
IPFS
GET
Hash String
SUBSCRIBER

Fig. 4.7 – File Transfer through IPFS network

STAGE 3: (Encryption using PGP)

While sending Hash through the IPFS network encryption is being done for allowing only the authorized users to decrypt the file and access the contents of it.PGP (Pretty Good Privacy) is used to encrypt the file using symmetric-key encryption and public-key encryption.

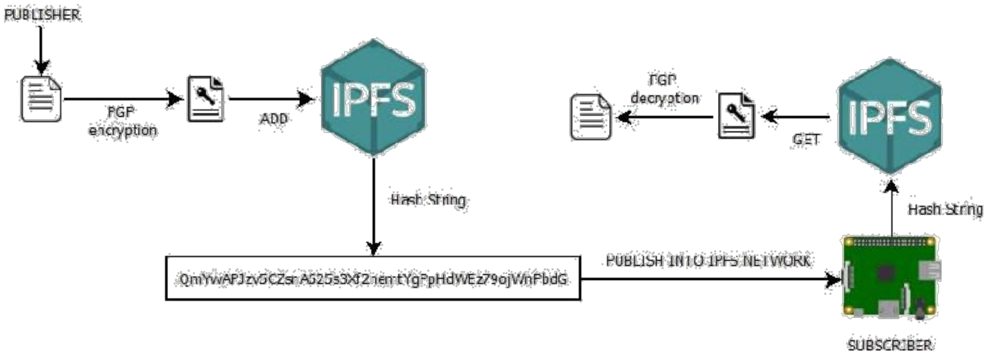


Fig. 4.8 – File Encryption using PGP

STAGE 4:

Changing the state of servo based on the data from the decrypted file at the subscriber.

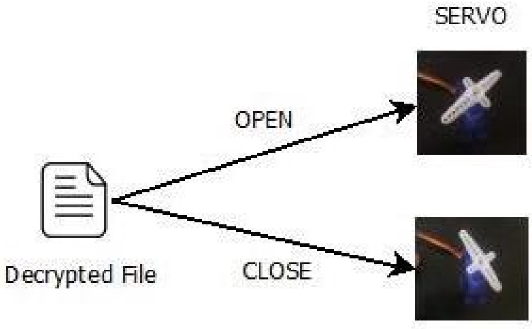


Fig. 4.9 – Setting Servo State from file

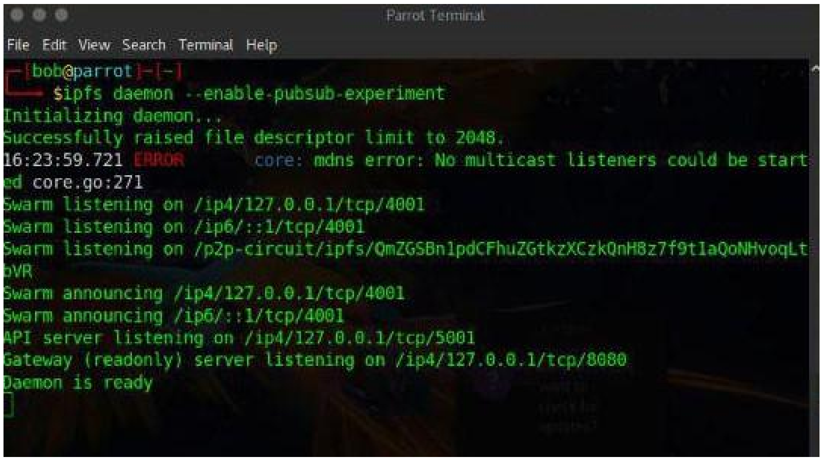
CHAPTER 5

RESULTS AND DISCUSSION

1.DAEMON RUNNING THROUGH CONSOLE

Starting the IPFS network by running the following command in Raspberry Pi

ipfs daemon –enable-pubsub-experiment

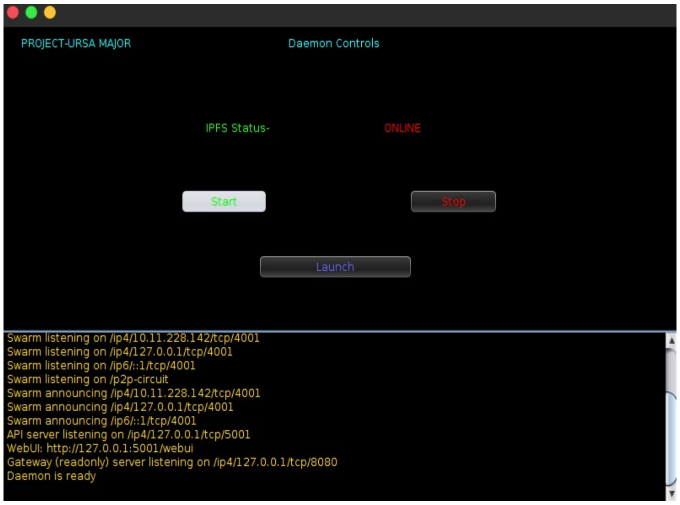


```
┌─[bob@parrot]─[~]
└──╼ $ipfs daemon --enable-pubsub-experiment
Initializing daemon...
Successfully raised file descriptor limit to 2048.
16:23:59.721 ERROR       core: mdns error: No multicast listeners could be start
ed core.go:271
Swarm listening on /ip4/127.0.0.1/tcp/4001
Swarm listening on /ip6/::1/tcp/4001
Swarm listening on /p2p-circuit/ipfs/QmZGSBn1pdCFhuZGtkzXCzkQnH8z7f9t1aQoNHvoqLt
bVR
Swarm announcing /ip4/127.0.0.1/tcp/4001
Swarm announcing /ip6/::1/tcp/4001
API server listening on /ip4/127.0.0.1/tcp/5001
Gateway (readonly) server listening on /ip4/127.0.0.1/tcp/8080
Daemon is ready
```

Fig. 5.1 – Daemon running in console

2.DAEMON RUNNING THROUGH THE JAVA APPLET (AT CLIENT)

Starting the IPFS network by running the following applet at Client.



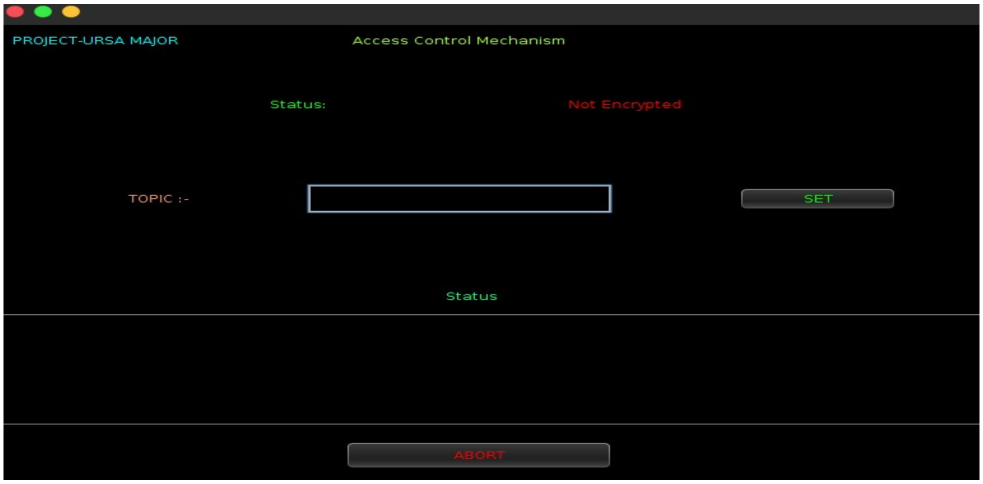
IPFS Status- ONLINE

```
Swarm listening on /ip4/10.11.228.142/tcp/4001
Swarm listening on /ip4/127.0.0.1/tcp/4001
Swarm listening on /ip6/::1/tcp/4001
Swarm listening on /p2p-circuit
Swarm announcing /ip4/10.11.228.142/tcp/4001
Swarm announcing /ip4/127.0.0.1/tcp/4001
Swarm announcing /ip6/::1/tcp/4001
API server listening on /ip4/127.0.0.1/tcp/5001
WebUI: http://127.0.0.1:5001/webui
Gateway (readonly) server listening on /ip4/127.0.0.1/tcp/8080
Daemon is ready
```

Fig. 5.2 – Daemon running through java applet

3. ACCESS CONTROL (AT CLIENT)

This is the module where communication and abstraction of data happens by giving the topic name.

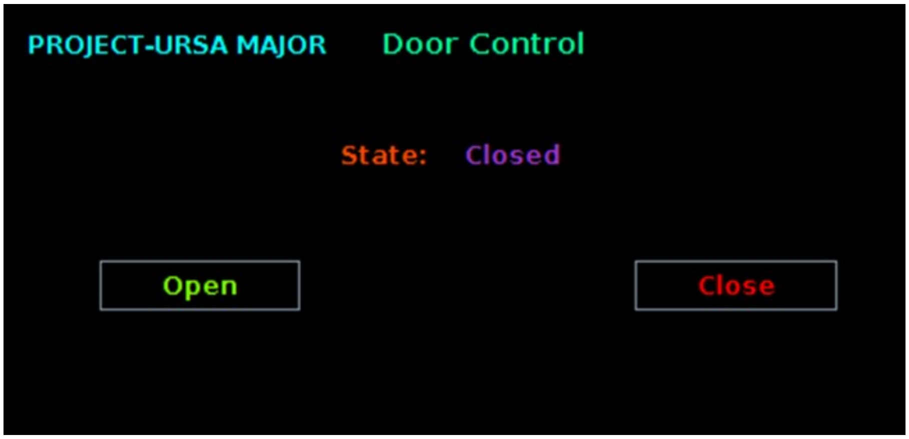


Status: Not Encrypted

Fig. 5.3 – Access control mechanism through java applet

4. CONTROLLING THE SERVO USING JAVA APPLET

Servo can be controlled by clicking on “OPEN” and “CLOSE” buttons

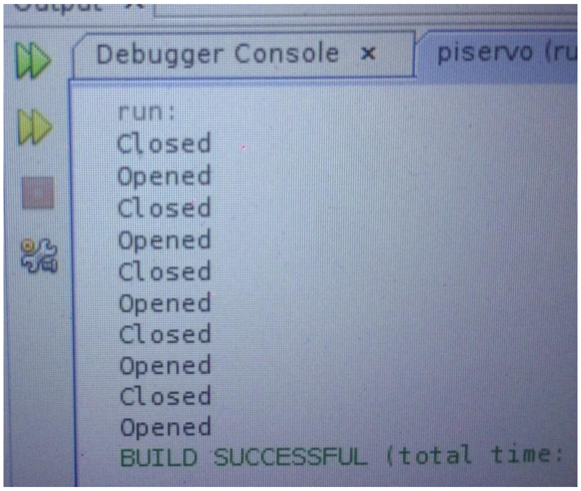


State: Closed

Fig. 5.4 – Controls for servo through java applet

5.SERVO RESULTS (AT NODE)

The Following Results are the values after the input is gone through decryption and then which is read by the Node to send signal to the servo to rotate.



```
run:
Closed
Opened
Closed
Opened
Closed
Opened
Closed
Opened
Closed
Opened
BUILD SUCCESSFUL (total time:
```

Fig. 5.5 – Servo results in console

6. SERVO STATES

The Below States are the motors open and close positions which act as door open/close controls.

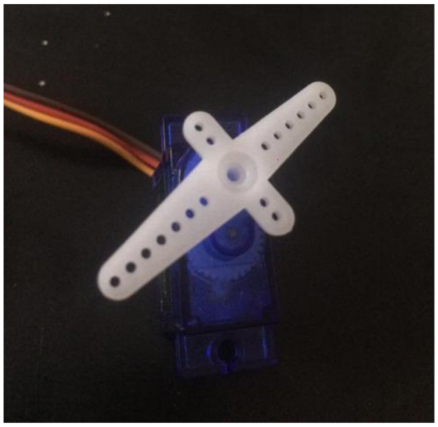


Fig. 5.6 – Servo in Open State

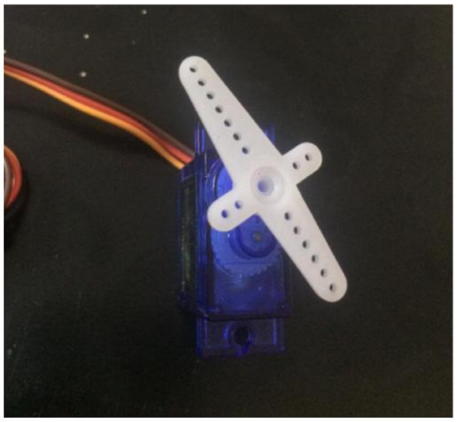


Fig. 5.7 – Servo in Closed State

7. BEACON NODE (AT CLIENT)

This is the module used to bring the channel up if there is a lot of disturbance in the network mainly for real time scenarios.

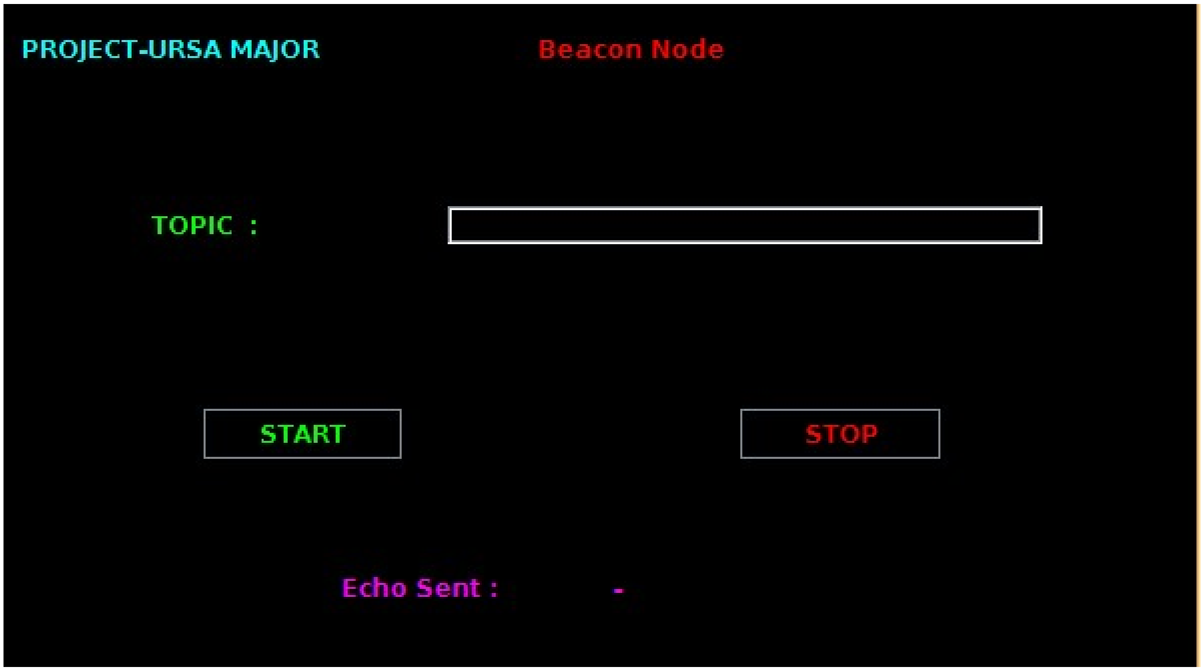


Fig. 5.8 – Beacon Node

CHAPTER 6

CONCLUSION

Inter Planetary File System indeed plays a major role in the field of D2D communication. It is not only keeping the data secure but also guarantees a total control of the device which can be accessible from any part of the world due to its distributed architecture. The project presents the use of IPFS in the field of Internet of things as a communication protocol which solves the problem of data accessibility and availability by scaling out centralized servers and cloud platforms. Which makes it reliable and inexpensive for users unlike protocols which uses brokers. So Future Enhancements' can be the following: -

- Improvised Filters to prevent various
- Tracker Node to Track Requests (i.e.: - Fire and Forget Service)
- Accessing Multiple Sensors via Same Channel using the above Model in Real Time.

Though the proposed models provide solution for establishing a secured connection from an end application to an IoT device without depending on third party service providers by using Peer-to-Peer architecture, it has some limitations.

- It requires more network resources.
- While finding path for the first time between end applications, it requires more time as it has to find the shortest path possible in real time.

CHAPTER 7

REFERENCES

[1] B. S. Adiga, P. Balamuralidhar, M. A. Rajan, R. Shastry, and V. L. Shivraj, "An Identity Based Encryption Using Elliptic Curve Cryptog- raphy for Secure M2M Communication," in Proceedings of the First International Conference on Security of Internet of Things, ser. SecurIT '12. ACM, 2012, pp. 68–74.

[2] D. D´ıaz Pardo de Vera, A´. Sigu¨enzaIzquierdo, J. BernatVercher, and L. A. Herna´ndez Go´mez, "A Ubiquitous sensor network platform for integrating smart devices into the semantic sensor web," vol. 14, no. 6. Multidisciplinary Digital Publishing Institute, 2014, pp. 10 725–10 752.

[3] H. S. Narman, M. S. Hossain, M. Atiquzzaman, and H. Shen. Scheduling internet of things applications in cloud computing. Annales des Te´le´communications, 72(1-2):79–93, 2017.

[4] Liu, Bin, et al. Blockchain based data integrity service framework for IoT data. Web Services (ICWS), 2017 IEEE International Conference on IEEE, 2017. https://doi.org/10.1109/ICWS.2017.54.

[5] "MQ Telemetry Transport," http://mqtt.org.

[6] Hunkeler, Hong Linh Truong, and Andy Stanford-Clark. MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. Communication systems software and middleware and workshops, 2008. Comsware2008. 3rd international conference on. IEEE, 2008.

[7] X. Wang, J. Zhang, E. Schooler, and M. Ion, "Performance evaluation of Attribute-Based Encryption: Toward data privacy in the IoT," in Communications (ICC), 2014 IEEE International Conference on, June 2014, pp. 725–730.

[8] P. Pal, G. Lauer, J. Khoury, N. Hoff, and J. Loyall, "P3S: A Privacy Preserving Publish-subscribe Middleware," in Proceedings of the13th International Middleware Conference, ser. Middleware '12, pp. 476– 495. https://doi.org/10.1007/978-3-642-35170-9_24.

[9] "ZigBee Alliance," http://www.zigbee.org.

[10] Agarwal, A., Singh, R., Gehlot, A., Gupta, G.,Choudhary, M.: IoT enabled home automation through nodered and MQTT. Int. J. Control Theor. Appl. (2017). ISSN 0974-5572

[11] Ali, Muhammad Salek, KoustabhDolui, and Fabio Antonelli: IoT data privacy via blockchains and IPFS Proceedings of the Seventh International Conference on the Internet of Things. ACM, 2017.

[12] Marco Conoscenti, Antonio Vetro and Juan C. D. Martin. “Blockchain for the Internet of Things: A Systematic Literature Review.” In Proceeding of The Third International Symposium on Internet of Things: Systems, Management and Security (IOTSMS-2016). https://doi.org/10.1109/AICCSA.2016.7945805.

[13] Benet, Juan. &IPFS-content addressed, versioned,P2P file system.1407.3561 (2014).

[14] OuaddahAafaf, Anas AbouElkalam, and AbdellahAitOuahman. “Towards a Novel Privacy-Preserving Access Control Model Based on Blockchain Technology in IoT.” Europe and MENA Cooperation Advances in Information and Communication Technologies. Springer International Publishing, 2017.

[15] Zyskind, Guy, Oz Nathan, and Alex Pentland. &Enigma: Decentralized computation platform with guaranteed privacy. & Xiv:1506.03471(2015).

[16] Loise Axon. 2015. Privacy-awareness in Blockchain- based PKI. Retrieved April 12, 2017from http://goo.gl/3Nv2oK

[17] Device Democracy: Saving the Future of the Internet of Things. Retrieved May 10, 2017 fromhttps://goo.gl/18Y16F

[18] Huckle, Steve, et AL,. Internet of things, blockchain and shared economy applications, Procedia computer science 98(2016): 461-466. https://doi.org/10.1016/j.procs.2016.09.074.

[19] Arvind P Jayan, Harini N, A Scheme to Enhance the Security of MQTT Protocol, International Journal of Pure and Applied Mathematics, Volume 119 No. 12 2018, 13975-13982.