JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 1193-1207
https://publishoa.com
ISSN: 1309-3452

Improving the Security and Privacy of Edge Devices for Internet of Things through Blockchain

R Radhakrishnan Unnithan[1], Manoj Yadav[2], Md. Aftab Alam[3]
[1]Department of Computer Science and Engineering, Lingaya's Vidyapeeth, Faridabad, India
[3]Department of Electronics and Communication Engineering, Lingaya's Vidyapeeth, Faridabad, India
[1]radhakrishnanpadanilam@gmail.com, [2]manoj200.yadav@gmail.com,
[3]mdaftabalam1986@gmail.com

Received 2022 April 02**; Revised** 2022 May 20**; Accepted** 2022 June 18**.**

Abstract

The Internet of Things (IoT) is a network of millions of computers and sensors that generate enormous amounts of data. It is sent to the cloud to be processed and computed; however, some computing paradigms are required near the edge devices due to latency, bandwidth, and storage constraints. Fog computing is a more advanced kind of cloud computing that extends some of its capabilities to end users. Within the Internet of Things, fog computing provides end-users with computational, storage, and network resources. One of the primary concerns of fog computing is authentication and safe data access between fog servers and fog nodes, as well as fog nodes and IoT users.Experiments were carried out to check communication between them, encrypting the data on transit, using symmetric and will a lightweight cryptographic algorithm. The experiments use PRESENT and AES algorithms to prove it. The system was breached to test the Ping of death and Smurf attack as DOS attack. The Sniffing for the password was done by Wireshark that also helped in the monitoring of the network. Thus, proving that encryption was a good solution for implementing security but the network was venerable to DOS and DDOS attack. In the objective to improving the security and privacy of edge device the research work opened avenue of using block chain a concept of implementing security by constructing two contract that were linked with each other. The concept helps to encrypt the data within the transaction while the sender and the receiver addresses are hashed which is the security benefits of the block chain.

Introduction

IoT is defined as network of connected device that has unique identification that follows an addressing protocol which has an embedded equipped technology that is able to intersect, sense, collect information and communicate about the ecosystem it is present in an around it. The reality of the ecosystem of IoT is how to use the data from sight that is automatically digitized an optimally to transform it into a business model Industry need. Not every connected device or nor their inherent capacities became a part of the IoT. The IoT is able to connect smart intelligent digital devices and humans. They link all edge devices with a protocol for communication that add data to it or capture data from it. This networking environment need to be secure.

Conventional IoT architecture [1] hastriode layers. The perception layer, network layer, and application are the three layers. I The physical layer, which has sensors for sensing and gathering information in its environment, is also the perception layer. It recognises physical parameters as well as other smart things in its surroundings. (ii) The network layer connects other smart things, such as servers, to other network devices. This layer is responsible for transmitting and processing sensor data. (iii) (iii)The application layer provides application-specific services. Many of them are placed at the IoT's edge. Smart home applications, health monitoring gadgets, and smart are some of the areas where they are used. Between the application layer and the network layer, there are support layers.

This architecture has been further modified so that the task of the IoT is still subdivided. Three layers, transport, process and business is inserted as shown in the table 1. The role of the perception and the application remain unchanged. The network layer is previewed as transport layer. It looks after the transit of data in the network. The different type of communication in this layer are Wi-Fi, RFID, NFC and many more. They also follow a protocol for communication. The process layer is the middleware layer. It stores data, analyses it as well as processing of the data that is received from the lower layer. The layer also provide service to the lower layer after managing the data that is send to it. The layer manages the database, cloud with the big data processing. The whole IoT system has various models which help in business and profit are at the business layer. This layer manages all the application and the security of the users. There are many models, but the model that includes seven IoT layers to understand the

TABLE 1 VARIOUS IOT ARCHITECTURE

	Three Layer		Five Layer		Seven Layer
3	The Application Layer	5	The Business Layer	7	People and Process Layer
2	The Network Layer	4	The Application layer	6	The Application layer
1	The Perception layer	3	The Processing layer	5	Data Analysis Layer
		2	The Transport layer	4	Data Ingestion Layer
		1	The Perception layer	3	Global Infrastructure Layer
				2	Connectivity/Edge Computing Layer
				1	The Things Layer

The need of IoT security is explained in two approaches. One is based on the properties it processes and the other through it layers. In IoT things show mobility due to the device that tends to hop from one network to another hence there is no stability in the network connection and there is no constant presence seen in the coverage in this environment. The cameras at junctions and other sensors that are used in the agricultural sector, automobiles, automotive industries that are publicly accessible and are expose to attack. It becomes critical for the IoT to manage users, services as well as gain trust among the automated measuring devices. The eco system where the device exist is heterogenous and belong to various manufactures. For the successful working these devices need to be integrated compatible and must be interoperable in each other environment while providing security. Due to the scalability of a large number of devices build up the network has to follow some protocol for an orderly working of the system. This will also be considered while setting up of the security mechanism for it. The PKIs that is public key Infrastructure show a centralizes approach [2]. The hierarchical and distributed approaches use pairwise generation of symmetric key exchange schemes but it is not able to scale-IoT network.

Structure of a blockchain

The blockchain is constructed to hold immutable blocks. Every block Bi>0does not change and is connected to the block before that is Bi−1. With the help of H(Bi−1) that is the hash of the Bi-1 is stored in the current. If there are change in any block it will have to reflect in the hash value stored store at its succession This will reflect in the creation of the new hash of the current. Therefore, having an avalanche effect. The first block B0 is the genesis block that does not have any parent and is the only block without a predecessor.In have block integrity it is signed digitally and the data is preserved.

A blockchain could be generalized and it stored arbitrary data. A block Bi is made up of:

- Hashed value Hi: It is hash value of block Bi-1: i.e., Hi: = H(Bi−1),
- Payload Pi: this is like the general data that occupies the block. It is decided on predefined pattern of transaction based on the type of BC.
- Signature Si: It is block information with digital signature, Si = Dsk(Hi | |Pi). This itself has a secret key of the owner of the block, sk‟. The verification is done by „pk‟ that is public key.

In public BC the new block is appended by the participant an on linking it to chain an immediately broadcasted. Any other participant on the blockchain can receivea new block and could consider it validity. This could be done by verifying signature, checking the hash of the pervious block, and checkingvalidity of payload. All copy confined to a node is broadcasted to peers in the chain after the new block is appended. If any error, it is marked and removed from the chain. If there are faulty participants the chain a consensus ondistributed network for Blockchains is attained by Byzantine Fault Tolerance (BFT), is a apt algorithm for an arbitrary number of peers. It should have at least a third of total number of participants [3]. Asynchronous networks used it successfully on blockchains. It workedpractically with about 1000 participants. It works on a asynchronous setup that has 1000 participants and incurred overhead on using the cryptographic algorithms.

A practical example is seen achieving consensus on millions of participants on asynchronous networks when there is on malicious node that controls many nodesit comprises of the Sybil attacks. The PoW is used for consensus among the Bitcoin BC. The validation is done on the bases of the amount of work done by the block that is seen by the time utilized on creating of it the node then varies the input of a hash function to get an output that has a certain pattern like having a number of leading zeros. This is a computationally very expensive problem., but by using the original it would be difficult for using cryptographic hash functions [4]. The process of creating a new block authorizing it and verifying it is called mining.

There is enough time and energy consuming forverifying transactions and appending new blocks.Therefore, there is reward for the nodes that has done the verification to do this node it is called mining rewards. Branching occurs when more than a single node makes more than one blockat a common time, A longer branch takes time and extensivework that is inclusive of validation. PoW is a consensus that does not allow attacker node on BC that sometime make node to spend twice on the same transaction or even forged information. This is double spending. A branch that is valid will have to tolerate at least 50% a malicious subset of nodes and must control at least 50% of the computing power in the network. The theoretical threshold for any general attack is only33% and the threshold has improved up to 50%.[5] Hence in this technology of BC not one part can be trusted and least half of the peers involved in creating and verifying hence consuming. The activities and the participant cannot falsely identify all the device addresses of the communication device and the key pair that use them are present when a transaction is created.

Related Work

Pradip Kumar Sharma et al. , 2020 [6] in this paper identifies that due to the growth of IoT device that was a surge in the consumption of power which ultimately would lead to increase in carbon emission due to its production. Therefore, it has become necessary that adapt to a green technology in IoT. Some of the ways that help to move towards green technology are to use cloud storge for data distribution, schedule the migration of data in a data center,creating an architecture such that distribution of energy can be dependent on the profile, demands and selection of the offloading from the devices depending in the data path, transferring the control at the edge. The authors combine the idea of blockchain technology used with IoT devices and suggest that renewable energy, certification of the devices that could be green oriented, using decentralization of data storage in clouds and using a framework that sustains green ecosystem must be implemented. **Bharat S Rawal and Yong Wang , 2019**,[7] notices that transporting private data on the public blockchain need more secrecy and management of credentials in the file sharing system. The authors propose a proxy re encryption (PRE) scheme that gives consent block by block. They were motivated by efficiency and privacy problem seen in Fork Problem, size of blockchain, data configuration time, Integration of cost and amount of energy consumed. The authors suggest a proxy re-encryption which allows semi trusted proxy to hide or convert cipher text from encrypted with one key to another.

Chunpeng Ge et al., 2019 [8] states that permissionless setting in Blockchain is when the mutually untrusting participants reaches a consensus on state of the distributed and decentralized ledger. The authors surveys on selected blockchain platform like Bitcoin, Ethereum and IoT which indicates accessibility, data structure recording and its size. In the case of IoTA logging becomes difficult and it is totally dependent coordinator. **Felix Franz et al., 2019** [9] is more subjective and theoretical to smart contracts. In the first part research is done to found out the requirements of the generator is simple to be programmed and has to be extendable without any change to the architecture for example RESTful APIs for project GeMARA, Etherparty Rocket, website Ethernaut. These places 22 challenges to handle venerability. Ethereum Improvement proposal and EIP-165 Interface detection were implemented to generate smart contracts.

G. Chander et al., 2019 [10] in this paper describe the FR chain that is usedSigning on multicast tree collectively so that verification and validation of a block is possible so that it can propagation as a valid block. They have considered crash fault, network fault and Byzantian faults which was implemented over 5000 nodes that where on 2 data centres 35 IBM SoftLayer cloud virtual machines and 30 Amazon EC2. The system model has CoSi combined with Schnorr multi-signature having a communication tree as a multicast protocol.

Work Methodology

The workflow follows a design process with four phases (as shown in Figure 1). The phases are exploration, design, experimentation implementation and evaluation.

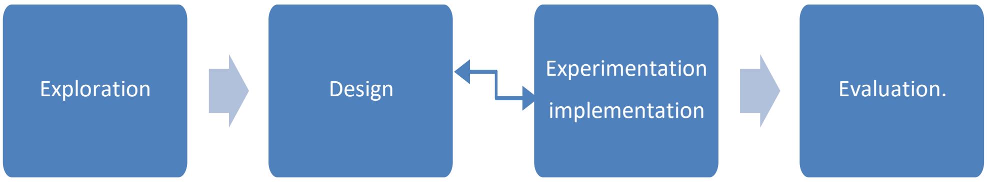


FIGURE 1: Work Methodology

The point of the **first phase**, the exploration phase, is to collect information and gain knowledge about the following areas sensors, communication and it types, IoT autonomy and architecture in relation with edge and fog Computing, IoT challenges, IoT vulnerability and risk in terms of Security threats and attacks, Blockchain application and its security advantages on IoT and Ethereum and its working. This helps in the study of the extensive work done in these areas. This is accomplished by reading about existing or proposed solutions to resembles the problems and scientific papers about similar problems or concepts. The study and gathering of literature and comparison exhibited lacunas in at security and at privacy of IoT. Blockchain, its features and application, Blockchain use in IOT security on different ecosystems and the reasons why blockchain is used and its implementation is analyzed.

The goal of the **second phase**, the design phase, is to come up with possible solutions to the problem. The information and knowledge from previous phase is helps to develop one or more possible solutions. The different solutions are then discussed from the case studies provided the research papers that fits most criteria and the time constraint, is chosen to implemented as a proof-of-concept in the next phase. The research method used is experimental   and is built on a prototype or a simulated frame of an IoT network with help of Raspberry Pi or similar SoC (System on Chip).

During the **third phase**, the experimental and implementation phase, the chosenprototype design is implemented. This is when most programming is done. The study of software and hardware as requirement prerequisite for the phase. In this phase a general prototype model of IoT which will create an environment to wireless connect THINGS with at the help of necessary hardware and software tools. If unforeseen problems arise or improvements are discovered while getting more hands-on experience it is allowed to change the design accordingly. This means that designing and implementation of experimental phase can overlap and repeat throughout the process.

In the last and **final phase**, the evaluation phase, the proof-of-concept solution is analyzed and discussed in regards to the objective of the project including the information and experience gathered during the entire process. According to the developments in the implementation of blockchain in the various areas of IT, finance, and others sectors. It is evident that it is successful in ensuring security in Bitcoin and financial industries data. This idea is implemented on the prototype of a smart home to ensure security and privacy of edge devices used in the prototype, hence and be extended to IoT device network and gateways. The use of smart contract and the idea of consensus to acknowledge every IoT atdevice level would give exhibit AAA in the blockchain. The steps use to develop a successful PoC

Step 1: Problem statement: Although there are many security measures that can be make the IoT edge devices secure, it is seen that 100% security is difficult to implement. A prototype model of IoT device that is include in the smart home is tested for breaches in environment that is controlled.

Step2: Stockholders‟: These are Edge devices in the IoT network are the component that are the stockholders in the system.

Step 3: PoC as a "solution": The Blockchain technology implemented on IoT edge device provides integrity, availability. It also provides AAA through smart contracts and mining. So, it an improved solution for edge device security.

Step 4: Detail setup of blockchain with set at the edge devices nodes with Ethereum blockchain are to be carried out. This also include the installation setup of the node that will dedicate miner to handle management activity of the chain

Step 5: Success measurements: This will enable to study how the data could be secure by using encryption beforeit is sent ontoedge devices network.

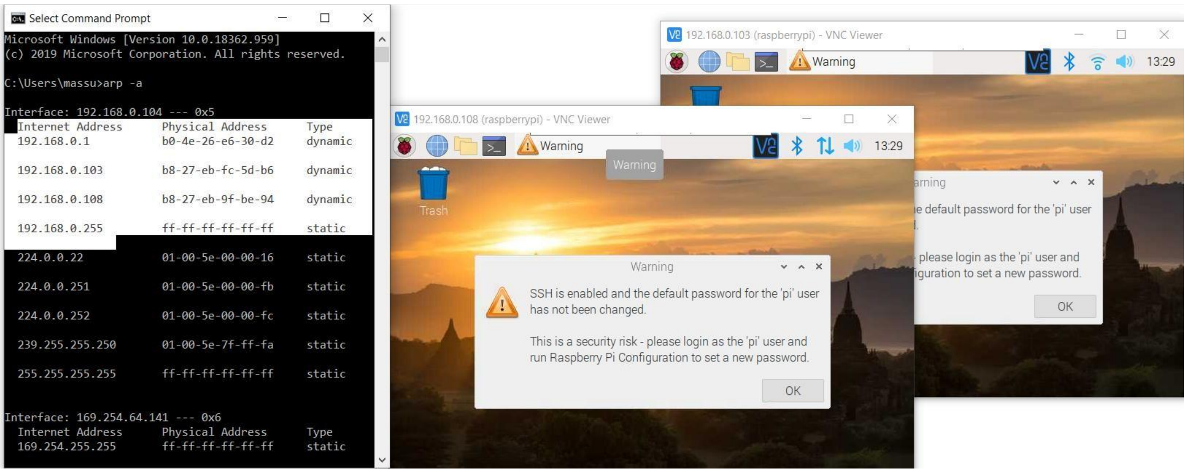
Experimental Setup

Ping of death uses the ping command. The ping command in general has the ability to check of the connection of the mention ip address is reciprocating. It measures the time it takes a packet to be send from the local host to the destination host and returns to source. There are many other functions of ping command like measures roundtrip time and record it. The ping must send 32 bytes of data to the destination that should reply in less than 1 ms and time to live is 128. This is the normal scenario but to make it an attack the send packet size could be large as 65500. Since that TCP/IP will have to fragment packetin smaller data. Chunks. Here the chunk itself is too large to be handled by the recipient that either it will crash or reboot or freeze/ hung.

From raspberry pi with ip address 192.168.0.104 the following command is sent ping 192.168.0.108 -t64 -s65500

-t -> continuously send data packet till the command is externally stopped

The unnatural flooding of the target computer doesn't affect the victim more, but if the command of the effect is seen this ping command has to be send from multiple Ip addresses



```
Microsoft Windows [Version 10.0.18362.959]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\massu>arp -a

Interface: 192.168.0.104 --- 0x5
  Internet Address      Physical Address      Type
  192.168.0.1           b0-4e-26-e6-30-d2     dynamic

  192.168.0.103         b8-27-eb-fc-5d-b6     dynamic

  192.168.0.108         b8-27-eb-9f-be-94     dynamic

  192.168.0.255         ff-ff-ff-ff-ff-ff     static

  224.0.0.22            01-00-5e-00-00-16     static

  224.0.0.251           01-00-5e-00-00-fb     static

  224.0.0.252           01-00-5e-00-00-fc     static

  239.255.255.250       01-00-5e-7f-ff-fa     static

  255.255.255.255       ff-ff-ff-ff-ff-ff     static


Interface: 169.254.64.141 --- 0x6
  Internet Address      Physical Address      Type
  169.254.255.255       ff-ff-ff-ff-ff-ff     static
```

192.168.0.103 (raspberrypi) - VNC Viewer
Warning
13:29
192.168.0.108 (raspberrypi) - VNC Viewer
Warning
13:29
Warning
Trash
Warning
OK
arning
OK

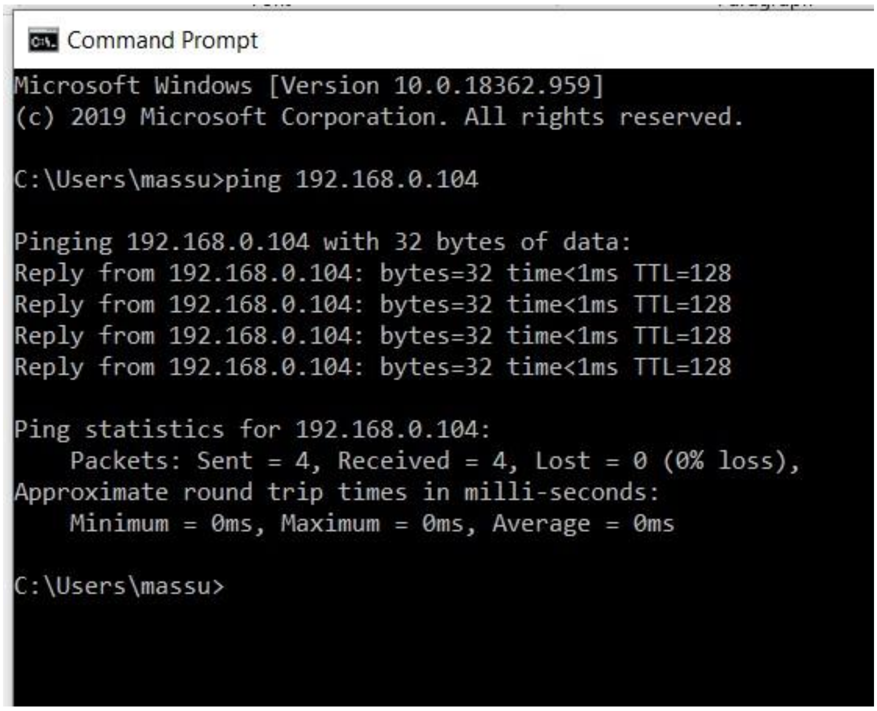
SSH is enabled and the default password for the 'pi' user has not been changed.

This is a security risk - please login as the 'pi' user and run Raspberry Pi Configuration to set a new password.

e default password for the 'pi' user

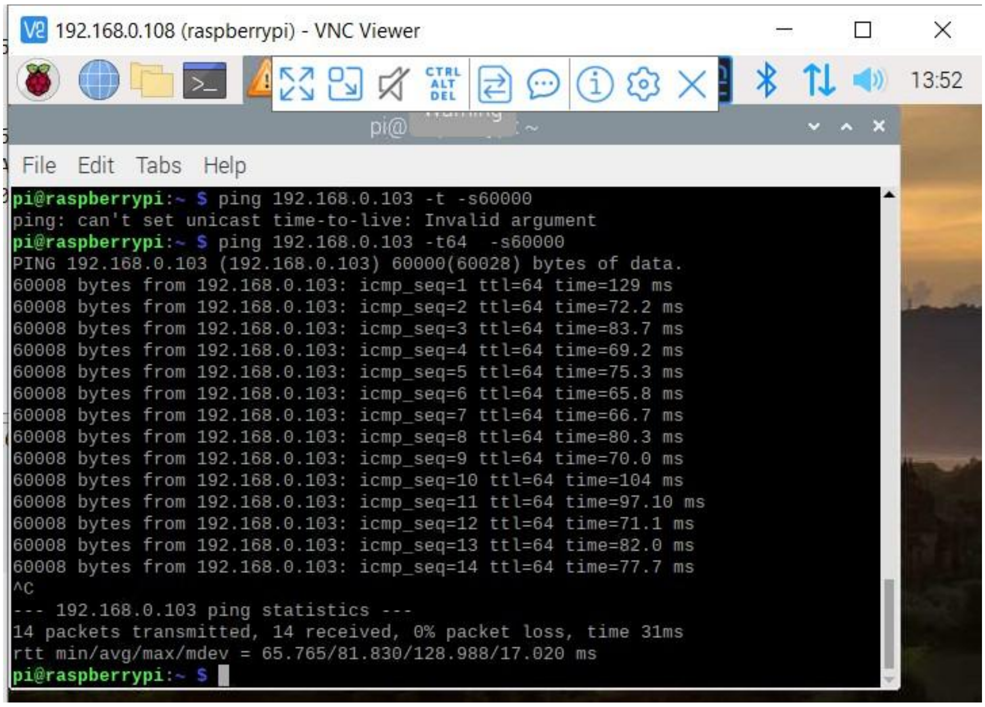
- please login as the 'pi' user and
iguration to set a new password.

FIGURE 2:Setup for Ping of Death



```
Microsoft Windows [Version 10.0.18362.959]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\massu>ping 192.168.0.104

Pinging 192.168.0.104 with 32 bytes of data:
Reply from 192.168.0.104: bytes=32 time<1ms TTL=128
Reply from 192.168.0.104: bytes=32 time<1ms TTL=128
Reply from 192.168.0.104: bytes=32 time<1ms TTL=128
Reply from 192.168.0.104: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.104:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\massu>
```

FIGURE 3: Ping Command



```
pi@raspberrypi:~ $ ping 192.168.0.103 -t -s60000
ping: can't set unicast time-to-live: Invalid argument
pi@raspberrypi:~ $ ping 192.168.0.103 -t64  -s60000
PING 192.168.0.103 (192.168.0.103) 60000(60028) bytes of data.
60008 bytes from 192.168.0.103: icmp_seq=1 ttl=64 time=129 ms
60008 bytes from 192.168.0.103: icmp_seq=2 ttl=64 time=72.2 ms
60008 bytes from 192.168.0.103: icmp_seq=3 ttl=64 time=83.7 ms
60008 bytes from 192.168.0.103: icmp_seq=4 ttl=64 time=69.2 ms
60008 bytes from 192.168.0.103: icmp_seq=5 ttl=64 time=75.3 ms
60008 bytes from 192.168.0.103: icmp_seq=6 ttl=64 time=65.8 ms
60008 bytes from 192.168.0.103: icmp_seq=7 ttl=64 time=66.7 ms
60008 bytes from 192.168.0.103: icmp_seq=8 ttl=64 time=80.3 ms
60008 bytes from 192.168.0.103: icmp_seq=9 ttl=64 time=70.0 ms
60008 bytes from 192.168.0.103: icmp_seq=10 ttl=64 time=104 ms
60008 bytes from 192.168.0.103: icmp_seq=11 ttl=64 time=97.10 ms
60008 bytes from 192.168.0.103: icmp_seq=12 ttl=64 time=71.1 ms
60008 bytes from 192.168.0.103: icmp_seq=13 ttl=64 time=82.0 ms
60008 bytes from 192.168.0.103: icmp_seq=14 ttl=64 time=77.7 ms
^C
--- 192.168.0.103 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 31ms
rtt min/avg/max/mdev = 65.765/81.830/128.988/17.020 ms
pi@raspberrypi:~ $
```

FIGURE 4: Ping of Death Attack

There are various options in the ping command that facilities this attack

-n determines the number of times the request has to be send to target

-l determines the size of data that each packet takes

-t determines the number of times to ping until the local host has timed out

Smurf attack

The smurf attack is another DOS type attack that flood the machine using broadcast ping message from which one can get address of the machine (this is snooping of address) the ICMP Internet control message protocol sends error massages to and from the machines in communication. There attacked forges the echo request of the ICMP packets with ip The ping of death was successfully carried out to case a denial-of-service attack. This attack was implemented to show the venerability of the IP address of Raspberry Pi which controlled the smart home. Hence blocking any legitimate activity of the attacked Raspberry Pi. This is accomplished by sending 65,535 bytes to the victim where at the data link has a highest frame size of 1560 byte over ethernet. The victim packet must split into several fragments and reassemble those fragments into a packet. In this attack the victim gets a packet greater than 65525 when reassembled and overflow of the memory buffer denying legitimate packets.

Creation of a Block chain

1. To test the accounts are created through go Ethereum

address of the targeted machine as the source and broadcast a request on the network. This make all machine on the network to reply to target which would be in hundreds and too large to listen to. In the DDos attack the attackersends a large set of requests from different Ips.

Results and Discussion

2. To create a smart contract for the smart home
3. To create smart contract with truffle and ganache

To install Ethereum on Rpi

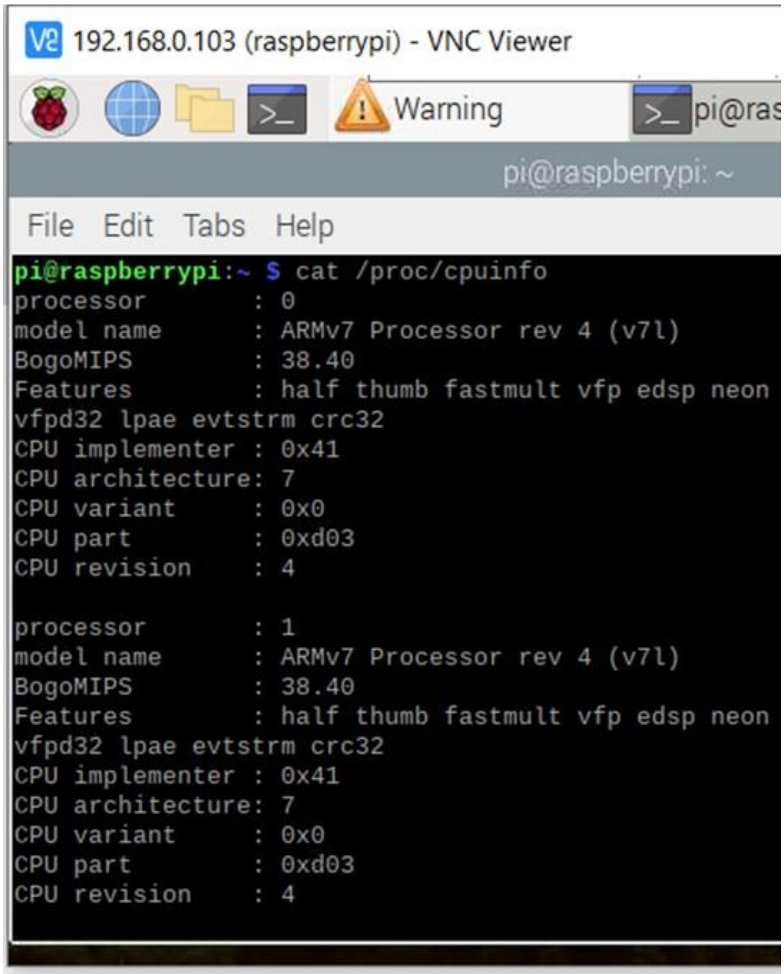
Check the Rpi CPU information by cat /proc/cpuinfo (figure 5)

Download the file from the net for the particular ARM processor

Pi $ wget https: //gethstore.blob.core.windows.net/builds/geth-linux-arm7-1.5.7da2a22c3.tar.gz pi$ cd geth-linux-arm7-1.5.7-da2a22c3 pi$ tar zxvf geth-linux-arm7-1. 5.7-da2a22c3.tar.gz pi$ sudo cp geth /usr/local/bin

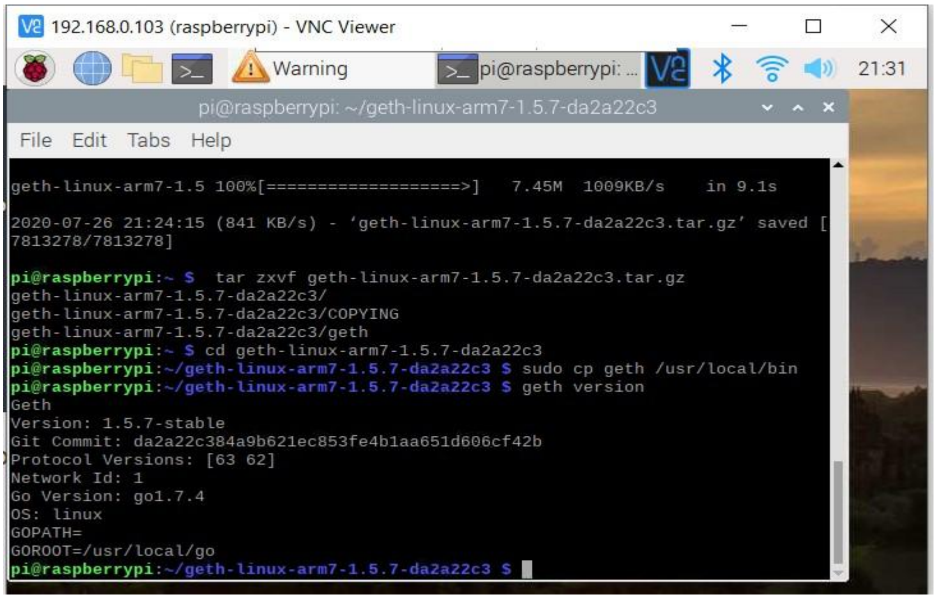
geth version figure 6 this show that Ethereum has been installed on the Rpi.

Similarly install Ethereum on the second Rpi.



```
pi@raspberrypi:~ $ cat /proc/cpuinfo
processor       : 0
model name      : ARMv7 Processor rev 4 (v7l)
BogoMIPS        : 38.40
Features        : half thumb fastmult vfp edsp neon
vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

processor       : 1
model name      : ARMv7 Processor rev 4 (v7l)
BogoMIPS        : 38.40
Features        : half thumb fastmult vfp edsp neon
vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4
```

Figure 5: CPU Information for Installation of RPI

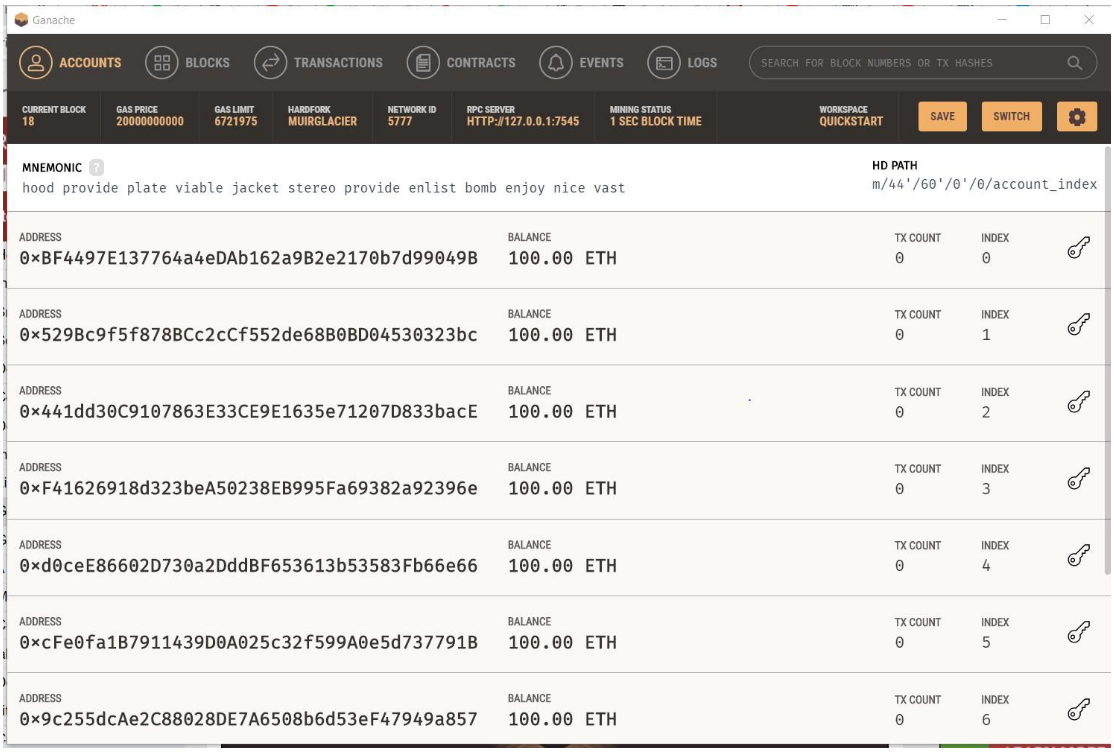

192.168.0.103 (raspberrypi) - VNC Viewer
Warning
pi@raspberrypi: ...
21:31
pi@raspberrypi: ~/geth-linux-arm7-1.5.7-da2a22c3
File
Edit
Tabs
Help

```
geth-linux-arm7-1.5 100%[===================>]   7.45M  1009KB/s    in 9.1s

2020-07-26 21:24:15 (841 KB/s) - ‘geth-linux-arm7-1.5.7-da2a22c3.tar.gz’ saved [
7813278/7813278]

pi@raspberrypi:~ $  tar zxvf geth-linux-arm7-1.5.7-da2a22c3.tar.gz
geth-linux-arm7-1.5.7-da2a22c3/
geth-linux-arm7-1.5.7-da2a22c3/COPYING
geth-linux-arm7-1.5.7-da2a22c3/geth
pi@raspberrypi:~ $ cd geth-linux-arm7-1.5.7-da2a22c3
pi@raspberrypi:~/geth-linux-arm7-1.5.7-da2a22c3 $ sudo cp geth /usr/local/bin
pi@raspberrypi:~/geth-linux-arm7-1.5.7-da2a22c3 $ geth version
Geth
Version: 1.5.7-stable
Git Commit: da2a22c384a9b621ec853fe4b1aa651d606cf42b
Protocol Versions: [63 62]
Network Id: 1
Go Version: go1.7.4
OS: linux
GOPATH=
GOROOT=/usr/local/go
pi@raspberrypi:~/geth-linux-arm7-1.5.7-da2a22c3 $ 
```

Figure 6: GETH Version

4.5.2 Tools to be Installed

1) Ganache. Change the setting to non-automatic and for one second mining figure 7.



CURRENT BLOCK 18 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MUIRGLACIER | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS 1 SEC BLOCK TIME | WORKSPACE QUICKSTART

MNEMONIC
hood provide plate viable jacket stereo provide enlist bomb enjoy nice vast

HD PATH
m/44'/60'/0'/0/account_index

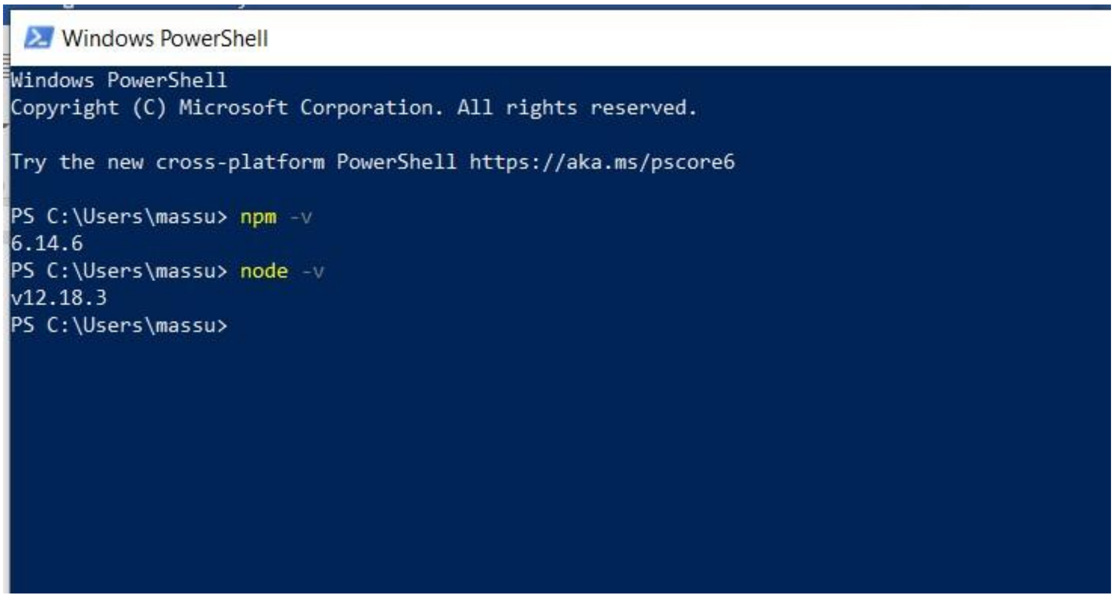
ADDRESS	BALANCE	TX COUNT	INDEX
0×BF4497E137764a4eDAb162a9B2e2170b7d99049B	100.00 ETH	0	0
0×529Bc9f5f878BCc2cCf552de68B0BD04530323bc	100.00 ETH	0	1
0×441dd30C9107863E33CE9E1635e71207D833bacE	100.00 ETH	0	2
0×F41626918d323beA50238EB995Fa69382a92396e	100.00 ETH	0	3
0×d0ceE86602D730a2DddBF653613b53583Fb66e66	100.00 ETH	0	4
0×cFe0fa1B7911439D0A025c32f599A0e5d737791B	100.00 ETH	0	5
0×9c255dcAe2C88028DE7A6508b6d53eF47949a857	100.00 ETH	0	6

FIGURE 7: Installation of Ganache

2) Install Node js . Use Powershell to check the version of npm and node

PS C:\user\massu>npm -v // 6.14.6

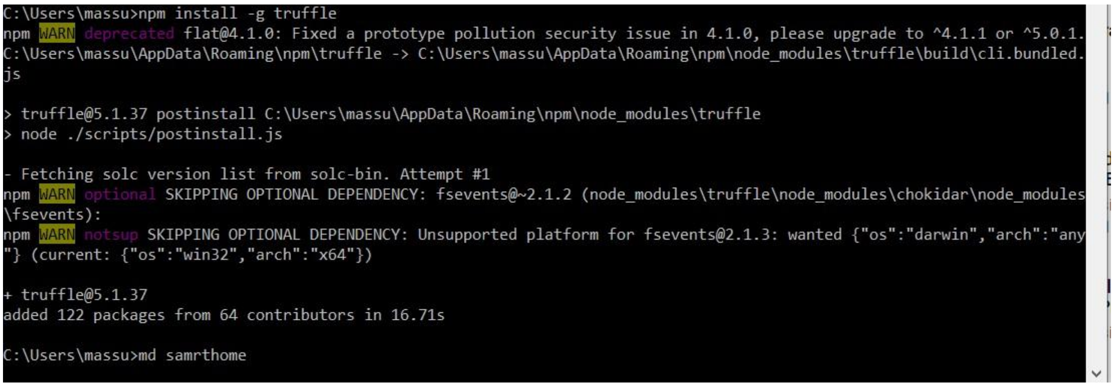
PS C:\user\massu>node -v // v12.18.3



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\massu> npm -v
6.14.6
PS C:\Users\massu> node -v
v12.18.3
PS C:\Users\massu>
```

FIGURE 8: Installation of Powershell to check NPM and Mode

3) Installing Truffle with the following command npm install -g truffle // g is for global



```
C:\Users\massu>npm install -g truffle
npm WARN deprecated flat@4.1.0: Fixed a prototype pollution security issue in 4.1.0, please upgrade to ^4.1.1 or ^5.0.1.
C:\Users\massu\AppData\Roaming\npm\truffle -> C:\Users\massu\AppData\Roaming\npm\node_modules\truffle\build\cli.bundled.
js

> truffle@5.1.37 postinstall C:\Users\massu\AppData\Roaming\npm\node_modules\truffle
> node ./scripts/postinstall.js

- Fetching solc version list from solc-bin. Attempt #1
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.1.2 (node_modules\truffle\node_modules\chokidar\node_modules
\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any
"} (current: {"os":"win32","arch":"x64"})

+ truffle@5.1.37
added 122 packages from 64 contributors in 16.71s

C:\Users\massu>md samrthome
```

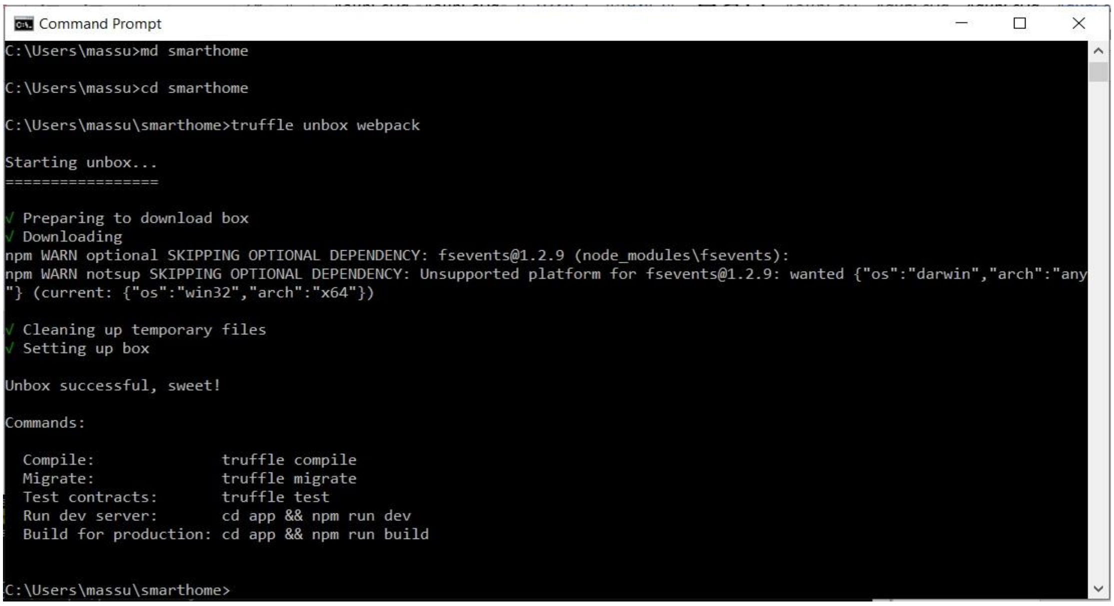
FIGURE 9: Installation of Truffle

4) To create project

C:\user\massu> md smarthome

Change the directory to it by cd smarthome

At the prompt C:\user\massu\smarthome> truffle unbox webpack.



```
C:\Users\massu>md smarthome

C:\Users\massu>cd smarthome

C:\Users\massu\smarthome>truffle unbox webpack

Starting unbox...
=================

√ Preparing to download box
√ Downloading
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any
"} (current: {"os":"win32","arch":"x64"})

√ Cleaning up temporary files
√ Setting up box

Unbox successful, sweet!

Commands:

  Compile:              truffle compile
  Migrate:              truffle migrate
  Test contracts:       truffle test
  Run dev server:       cd app && npm run dev
  Build for production: cd app && npm run build


C:\Users\massu\smarthome>
```

FIGURE 10: Unboxing of Truffle Webpack

Solidity.

Solidity is a high-level contract-oriented language. It is used to write smart contracts. It is developed by contributors of Ethereum Blockchain Platform. It helps on designing as well as the implementing smart contracts inside the Ethereum Virtual Machine. It also helps in the other blockchain development platforms. Using it the developers write applications which require business logic with smart contracts in it. Solidity is designed like JavaScript syntax and it is easy for web developers to implement it. The main functionalities of Solidity are:

- It support multiple inheritances having C3 linearization.
- It supports of state variables, objects, data types, and other functions that make programming easy.
- In contracts complex member variables like structs and hierarchical mapping can be used.

- several type-safe functions within a contract can be used with binary interfaces
- Miner initialization of the device is done by sharing a key and then making a transaction through the agreement contract
- Miner applies the agreement policy in the block created,
- the updated device agreement policy with the users, is to be maintained in the local store
Device Transaction
- When a device is added to the smart home network the agreement policy is generated only then the device can perform a transaction.
- If light button has to activated the data request has been received by the sensor to automatically turn on the lights. On checking the status action is performed
- If the device needs to communicate with another device the Miner shares a key to the devices such that these devices can communicate with each other
- The miner with the help of the agreement policy asks permission from the owner to sharing and then share with the devices, hence the device happens to communicate with each other
- The devices tend to communicate with each other till the key is valid, once the shared key is invalid the permission is denied and the communication is stopped by the miner
- All transaction are stored on the local store
- For anytransaction in the smart home the devices useit granted a key by the miner which the device will have to make a request. Thus, each device needs to be authenticated by the shared key to store locally. The device uses the key and the local storage shared key access or store the contents. Therefore, device uses the shared key and is able to collect data on the local storage system.
- Theneed for authorization comes each time the device has to store the data. The miner will extract the latest block‟s block number and its hash from the blockchain. This happened in the store transaction where the hash and the data are sent.
- The access transactions are when the miner receives the request for data in the local store system. If the data is present then it sends data to the participant how asked for it or it sent the latest block Number or the hash of the requester. The data storage must be an individual device to prevent security attacks during linking.
- The monitor transaction or a viewing a transaction is when the miner passed current data of a device to another device that requested for it.
- To keep the check on the requesting device receiving the data in time period can be set of t second which the miner will abort transaction if the data is not sent in the speculated time , the truncation gets aborted and stored as incomplete in chain.

4.7 Security in the Blockchain proposed system

- The proposed system exhibits confidentiality when the individual authorized user and read the data. Thus, getting the users authorized by the miners gives confidentiality.
- The proposed system exhibits Integrity when the message passing from the source to the destination does not change
- The proposed system exhibits availability was the services and data are available when the entity requires them. There is a need to improve the protection on availability of the data for malicious attacks like request for anonymous sources This is done by limiting the acceptance of transactions and given access to only to the one who have shared keys. This is done by the miners as well as authorising and forwarding the request.
- As the prototype shows hierarchical form device are not directly accessible hence there is no chance of the it been attacked by the malicious malware that could be installed by that an attacker. Therefore, fouling the attempts of a DDOS attack
- As all transaction and activities are checked and traced by the miners

- If the system is attacked any traffic moving towards has to be authorized by the miner after checking the agreement.
- The miners are the ones that create the ledgers when they are links set up between devices with different Keys. But these devices have their data shared as well as stored by a unique key this is used for each transaction by the miner. therefore, avoiding linking attack
- To avoid possible attacks, there should a limitation in minutes for the acceptance of data periodically, when time expires the connection should be terminated.
- It is noticed that when the transactions are executed between two device the send and the receiver"s identity are hashed. But the data is open. In the block chain currently, there is no idea of data hiding on the chain. Hence it is very venerable to loss on confidentiality. To avoid this there a 4 suggestion that can be implemented.

a) Make the data variant private
b) Implement encryption algorithms on the data before sending it across. The type of algorithm could be preferable lightweight as it has to be implemented a low-level processor / controller or IoT. One can send the key necessary for decryption on the receiver device that initiates the action
c) In Solidity there are a few cryptographic function that can be use for hashing the data [60] and ABI is Application Binary Interface (ABI) that has a method to communicate between contracts and outside elements of Ethereum. Data encoding is not done automatically and has to have a schema to decode. The interface functions in contract is a strong datatypes that is static at compiled.
d) Simple encryption with transportation and substation techniques can be used. But this will have to be written as user defined function as it will be efficient for the IoT edge device requirement.

Conclusion and Future Scope

This research work done is to give a better upgraded solution for security and privacy. The work successfully describes the study of a Raspberry Pi, making it an edge device on a network and using it creating a prototype of smart home is the basis. The work progress to use security techniques of encryption especially lightweight cryptography that is recommended for edge devices as solution to security. DOS attack is currently the easiest and most frequently use threat on IoT, some of which are implemented in the work. With the help of Proof of concept and the study of Ethereum blockchain a better solution is recommended for security and privacy of edge devices. To set up the Rpi as the smart home the WebIOPi was used the JavaScript was use to carry the button to light the LED and a command and the Python program lit the LED. This proved that command could be passed for a web front end to a RPi that represented the edge device of a Smart Home. The next set of experiments successfully implement the ping of death attack and the smurf attack. As future scope of the research implementation of the private blockchain will make the system more reliable.

References

[1] https://www.internetsociety.org/resources/doc/2019/trust-opportunityexploring-consumer-attitudes-to-iot/.

[2] M. Farooq, M. Waseem, A. Khairi, and S. Mazhar, "A Critical Analysis on the Security Concerns of Internet of Things (IoT)," Perception, vol. 111, 2015.]. 106 M. Leo, F. Battisti, M. Carli, and A. Neri, "A federated architecture approach for

Internet of Things security," in Euro Med Telco Conference (EMTC), 1-5, 2014.

[3] https://searchsecurity.techtarget.com/definition/cryptography.

[4] https://www.slideshare.net/rjain51/iot-ad14.

[5] https://www.slideshare.net/ashutoshb418/challenges-and-application-ofinternet-of-things.

[6] Pradip Kumar Sharma, Neeraj Kumar, and Jong Hyuk Park ,2020, "Blockchain Technology Toward Green IoT: Opportunities and Challenges", 0890-8044/19, May 06,2020 IEEE.

[7] Bharat S Rawal and Yong Wang ,2019, ―Splitting a PRE-scheme on Private Blockchain‖, published at 2019 IEEE Canadian Conference of Electrical and Computer Engineering, ISBN:978-1-7281-0319-8, ISSN:2576-7046 16 Bogdanov et al. at CHES 2007 [25 B. Collard and F.-X. Standaert, ―A Statistical Saturation Attack against the Block Cipher PRESENT.

[8] Chunpeng Ge, Siwei Sun, and Pawel Szalachowski, 2019―PermissionlessBlockchains and Secure Logging‖ published at 2019 IEEE International Conference on Blockchain and Cryptocurrency, arXiv:1903.03954.

[9] Felix Franz,Tobias Fertig, Andreas E. Schütz, Henry Vu,2019 ‖Towards Human-readable Smart Contracts‖ published at 2019 IEEE International Conference on Blockchain and Cryptocurrency, ISBN:978-1-7281-1328-9, DOI:10.1109/BLOC.2019.8751309

[10] G. Chander, Pralhad Deshpande, Sandip Chakraborty, 2019 ‖A Fault Resilient Consensus Protocol for Large Permissioned Blockchain Networks‖, 2019 IEEE International Conference on Blockchain and Cryptocurrency, ISBN:978-1-7281-1328-9, DOI:10.1109/BLOC.2019.8751439.