Augmenting smart home network security using blockchain technology

Utkarsh Saxena*, J.S. Sodhi and
Rajneesh Tanwar

Amity University,
Sector 125, Noida, India
Email: Utkarshsaxena115@gmail.com
Email: jssodhi@akcgroup.in
Email: rtanwar@akcds.in
*Corresponding author

Abstract: The idea of smart home existed from 1970's onwards but has come into the knowledge of researchers and data scientist due to the development in the domain of internet of things (IoT) (Kopetz, 2011), but it still suffers from privacy and security vulnerabilities. Conventional security policies or approaches are not applicable to IoT, mainly due to its decentralised topology and the resource constraints of the majority of its devices (Dorri et al., 2017a). This Paper presents an innovative, novel and decentralised approach that can be used to augment the existing security architecture of a smart home network. Our scheme guarantees both trustworthiness and user privacy preservations.

Keywords: smart home; internet of things; security; blockchain; decentralised approach; user privacy; encryption; data structures; peer-to-peer computing; data sharing; proof-of-data.

Reference to this paper should be made as follows: Saxena, U., Sodhi, J.S. and Tanwar, R. (2020) 'Augmenting smart home network security using blockchain technology', *Int. J. Electronic Security and Digital Forensics*, Vol. 12, No. 1, pp.99–117.

Biographical notes: Utkarsh Saxena is a doctoral student and Assistant Manager at Amity University, where He is pursuing a Doctorate of Philosophy (PhD) in Computer Science and Engineering. His areas of interests are networks security, and algorithm designing. He received his Bachelor of Technology in Computer Science and Engineering from Uttar Pradesh Technical University and plans on pursuing a Doctorate of Philosophy in Network Security, with a concentration in smart home security architecture.

J.S. Sodhi is a Sr. Vice President (IT) in Amity University. His areas of interests are networks security, and network infrastructure management. He received his Bachelor of Technology in Computer Science and Engineering from GB Pant Technical University. He has managed more than 100+ IT companies as an advisor.

Rajneesh Tanwar is an Information Security Analyst – Research and Development team at Amity University, where he research over the solution in security aspects by utilising new and advanced technologies. He received his Master of Technology (2017) for Computer Networks and Information Security from Amity University, Noida and Bachelor of Technology (2015) for

Information Technology from G.G.S. Indraprastha University, New Delhi. He had contributed in European Government Project in Germany (2017) as his Master of Technology thesis where he proved work on data protection in resource-limited environment.

1 Introduction

The internet of things (IoT) is one of the most significant, emerging and disruptive technologies of this century. Here Things are embedded cyber-physical system, which means that they are not the computer itself but they contain a computer inside it through which it can be controlled, monitored and accessed.

However, the increasingly invisible, dense and pervasive collection, processing and dissemination of data in midst of people private lives gives rise to serious security and privacy concerns (Henrik et al., 2015).

In recent years, an increasing number of security-based solutions (SBS) have been released, mostly because of the rapid expansion of the mobile device market (Henrik et al., 2015). These approaches take advantage of existing securities policies or algorithm to provide a secure architecture for a smart home network.

A proof of transmission is a digital certificate that ensures transmission of the message, at a certain time. Different security schemes have been proposed which are either infrastructure-dependent or infrastructure-independent. It is worth noting that most schemes are centralised i.e., they rely on the central server for communication.

With the objective of achieving a secure architecture, which at the same time provides verification of devices as well as messages. We have designed a completely decentralised, infrastructure independent scheme for secure data transmission in a smart home network.

Our proof of data transmission is based on blockchain technology. Blockchain is shared peer to peer distributed ledger (distributed database). It is a peer to peer transaction management system without an intermediary (https://www.tcs.com/content/dam/tcs/pdf/technologies/internet-of-things/abstract/Blockchain%20for%20the%20IoT.pdf). The transactions are verified by a network of nodes and recorded in a public distributed ledger called blockchain.

1.1 Reason for adopting blockchain for smart home

The blockchain contains intrinsic security mechanism against various network threats and due to this reason we adopted this technology for the device to device communication in a smart home network.

1 *Trust Building:* since blockchain communication works between the trusted nodes i.e., only trusted nodes can make a transaction. So it ensures that only trusted device can make communication in a network.

2 *Cost-effectiveness:* it reduces processing overhead and eliminates the 'middle man' (IoT gateways). Since IoT devices are low power (Mendez et al., 2017).

3 *Accelerate data exchange:* since no ‘middle man (IoT gateways) are present in between the communication, it decreases information exchange and processing time.

4 *Scalable:* since blockchain technology ensures that it works efficiently even on connecting large number of devices and thus it is scalable in nature. Since in a smart home network, many devices are connected to each other, the blockchain technology helps to provide low latency, high throughput, querying, permissions and decentralised control.

The rest of the paper is organised as follows. In Section 2 we describe the related work. In Section 3, we illustrate our proof-of-data scheme. In Section 4, we analyse the proposed schemes in terms of robustness against various kinds of attacks. In Section 5, we illustrate a preliminary performance evaluation of the proposed scheme. Finally, we conclude the paper with a summary of the achieved result and outline for the future work in Section 6.

2 Related work

Dorri et al. (2017b) describe a security approach for smart home network using blockchain technology. They extended proof of work (POW) concepts in order to reduce processing overhead occurred due to miners. This scheme divides the smart home into three core components:

a devices

b cloud storage

c blockchain.

They used shared overlay network through which syncing of blockchain occurs. But deploying a shared overlay network using cloud storage is costly one.

According to Christidis and Devetsikiotis (2016), IoT device were manipulated by etherium blockchain platform with smart contracts for tracking meter and setting policies to control on and off our conditioner and light bulbs in order to save energy.

Dorri et al. (2017c) proposed a combination of private and public blockchain. Private blockchain was employed to handle data flow in smart home system (SHS), whereas public blockchain was to manage data flow over cloud storage. Their approach comprised of three tiers: SHS, overlay network, cloud storage

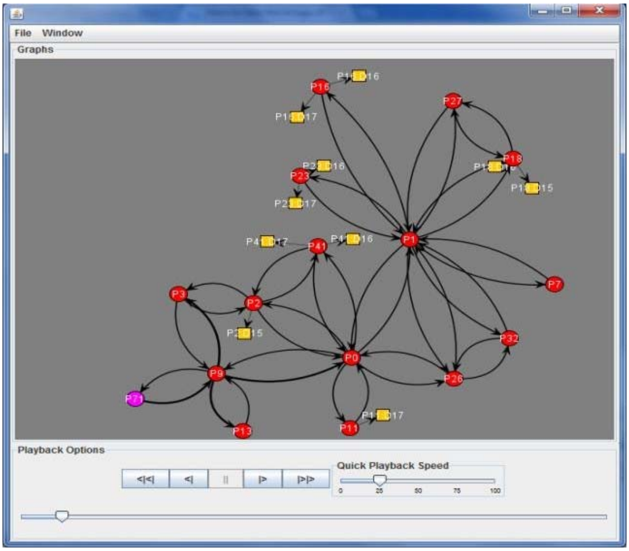
Stanciu (2017) proposed hyperledger fabric as the blockchain solution, where function blocks are to be implemented as smart contracts on a supervisor level. Edge nodes perform on the executive level. It uses Kubernets plate form which is used for orchestrating the execution of containers across the edge resources.

An approach for developing a campus-wide sensor network using commodity single board computers (Raspberry pi) was presented in Henstchel et al. (2016). Edge computation is made of per-node event triggers that are defined and monitored on each sensor device directly. Further, computation tasks might include local aggregation of data such as map that reduce per-node user’s queries executing on device.

3 Architecture

Our approach considers smart home network as a peer to peer network in which each device is connected to other devices in a peer and thus forms a peer group for communication. For analysing peer to peer architecture we use JUNG framework which is a network visualise which uses XML files to display graphs of peer to peer network and contains the log event. The simulator image is shown in Figure 1.

Figure 1 Peer to peer network simulation using Jung blockchain construction (see online version for colours)



We use the undirected and unweighted graph G = (V, E) to characterise the network connectivity structure of a smart home network, where V is the set of nodes (devices) with size n, E is the set of edges (connections) with size m.

Equivalently, the graph can be represented by an n-n binary symmetric adjacency matrix *A*, where $A_{ij} = 1$ if there is an edge between nodes *i* and *j*, otherwise $A_{ij} = 0$ (Chen, 2014).

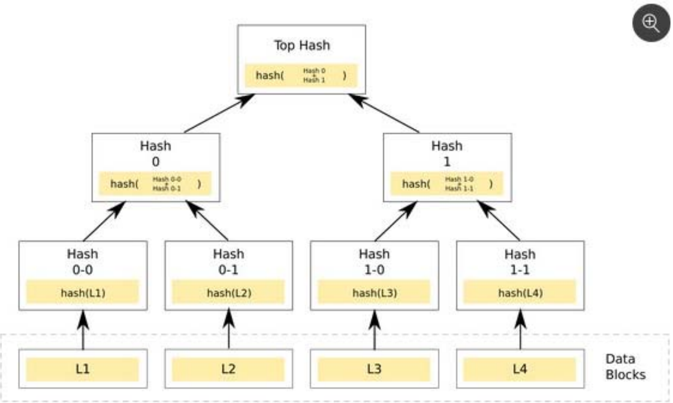
3.1 Merkle tree

A merkle tree is a way of hashing a larger chunk of data into a single hash. It is a tree structure in which each leaf node is a hash of a block of data, and each non-leaf node is a hash of its children (https://brilliant.org/wiki/merkle-tree/).

Merkle tree is used in a distributed system for effective data verification. They are efficient because they uses hashes instead of full files (https://brilliant.org/wiki/merkle-tree/).

Merkle trees are typically implemented as binary trees, as shown in the following image. However, a merkle tree can be created as an n-nary tree, with n children per node (https://brilliant.org/wiki/merkle-tree/).

Figure 2 Binary merkle tree[13] (see online version for colours)



Source: Stanciu (2017)

In this image (Figure 2), we see an input of data broken up into blocks labelled L1 through L4. Each of these blocks are hashed using some hash function. Then each pair of nodes are recursively hashed until we reach the root node, which is a hash of all nodes below it. In this image, we see an input of data broken up into blocks labelled L1 through L4 (https://brilliant.org/wiki/merkle-tree/).

Complexity	*Average*	*Worst*
Space	O(n)	O(n)
Search	$O(\log_2(n))$	$O(\log_k(n))$

However, a very important aspect of merkle tree is in the synchronisation of data.

Synchronisation	$O(\log_2(n))$	$O(n)$

3.2 Blockchain construction

We have adopted the blockchain technology to show networked nodes with the capability to verify and store transmitted messages, not requiring a centralised supernode that oversees sensitive data of other nodes. In our approach, recent valid data are recorded into blocks, which are then added to the end of the chain and, once confirmed by consensus, they cannot be changed, as shown in Figure 2.

Similarly to the solution proposed by Zaghal (2016), peers can communicate with near nodes through any short-range communication technology, such as Bluetooth, Bluetooth SMART or ZigBee, and they periodically use these interfaces to broadcast proof-of-data requests and responses to their neighbours, as illustrated in Figure3.

Figure 3 Proof-of-data scheme in blockchain (see online version for colours)

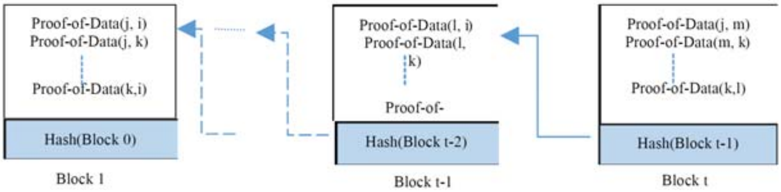


Figure 4 Verification of message in our approach

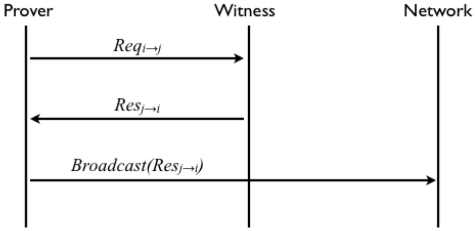
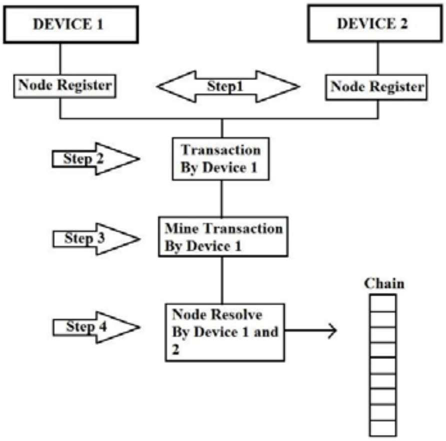


Figure 5 Normal work flow of blockchain communication



3.3 Experimental setup

For the experimental demonstration, we setup a smart home environment using ARM-based devices. We choose ARM-based devices because the devices exist in a smart home resemble to the ARM-based device. Since ARM Processors is low cost and low storage and limited processing capability devices it suits our experiment.

We choose the following components for our experiment

1 Raspberry pi (Model3B)

2 Wi-Fi adapter

3 Jessie OS (Raspberry OS image)

4 Python3.6+

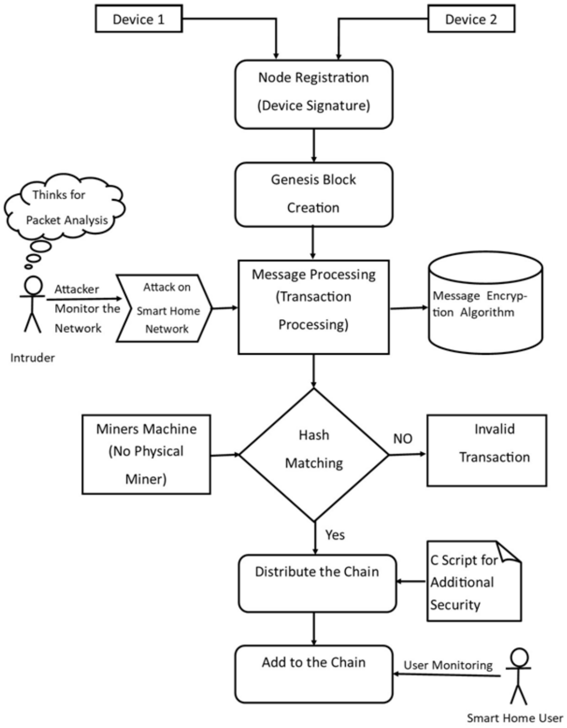
5 Blockchain program

6 Base64 encoding.

4 Proposed scheme work flow

The working of our approach is demonstrated in the flow chart, it is more similar to the previous workflow except from the additional security aspects which we have added in our approach. The detailed description is shown in Figure 6.

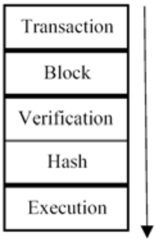
Figure 6 Proposed work flow


Device 1
Device 2
Node Registration (Device Signature)
Genesis Block Creation
Thinks for Packet Analysis
Attacker Monitor the Network
Intruder
Attack on Smart Home Network
Message Processing (Transaction Processing)
Message Encryption Algorithm
Miners Machine (No Physical Miner)
Hash Matching
NO
Invalid Transaction
Yes
Distribute the Chain
C Script for Additional Security
Add to the Chain
User Monitoring
Smart Home User

The whole process is divided into five phases, and these phases are defined as follows.

- *Transaction:* two devices, A and B decide to exchange a message and initiate the communication.
- *Block:* the message is packaged with other pending messages thereby creating a block. The block is sent to the blockchain system's network of participating computers.
- *Verification:* the participating devices (miners) evaluate the message through mathematical calculations which are used to determine whether the message sent is valid, based on the agreed upon rules, when consensus has been achieved, typically among 51% of participating computers (Conti et al., 2017).
- *Hash:* each verified block of transactions is time – stamped with a cryptographic hash. Each block also contains a reference to the previous block's hash, thus creating a 'chain' of records that cannot be falsified except by convincing participating computers that the tampered data in one block and in all prior block is true.
- *Execution:* finally the message passes from one device to another.

Figure 7 Message passing levels



Working:

1 Initially device registers itself using its device signature which will act as a private key of that device which will be used when hat device transmits the message.

2 After the registration process, the genesis block will be created. Since it is the first block which is formed when the communication process starts using blockchain.

3 Whenever a device sends a message, it sends a message using its private key. This message gets distributed over the network.

4 The miner's computers or devices that are already present in the network validate the message, whether it is been sent by an authentic device or injected through any other external party.

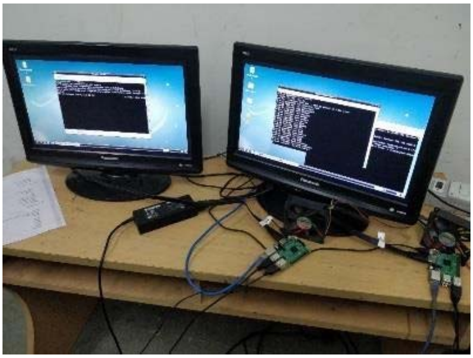
5 The miners will calculate its Hash, by solving some complex mathematical calculations.

6 If the hash calculated by the mining device verifies the message then this message will get added in the chain of existing block and thus it forms a blockchain

7 Since we deploy an additional C script which will determine the existence of malicious command in the message.

8 If it contains any malicious code then it will discard the transaction and that message will not get added in the blockchain. And hence that device will not participate in the communication process.

Figure 8 Raspberry pi setup for device communication (see online version for colours)



Steps to form block chain:

1 Install python 3.6 because blockchain will run only on the virtual environment created by this version of python.

2 Update all other packages like gcc, etc.

3 Finally, form blockschain using the created virtual environment.

Steps after the creation of block chain:

1 View Chain:

Browser: http://**IP_Address:Port_No**/chain

Over Terminal: curl-X GET-H "Content-Type: application/json" "http://**IP_Address:Port_No**/chain"

2 For mining:

Browser: http://**IP_Address:Port_No**/mine

Over terminal: curl -X GET -H "Content-Type: application/json" "http://**IP_Address:Port_No**/chain"

3 For transaction: (can't be done using the browser. User have to install POSTMAN)

POSTMAN:

1 Select POSTmethod

2 Enter the message in "body tab by selectingjson".

3 Message: {"sender": "abc", "recipient":"xyz", "amount":"hi!"}

4 Enter http://**IPADDRESS:PORT_NUMBER**/transactions/new

Terminal:

curl -X POST -H “Content-Type: application/json” -d ‘{ “sender”: “abc”, “recipient”: “xyz”, “amount”: “hi!” }’ “http://**IP_Address:Port_No”/**transactions/new

Two make communication between two nodes then before transaction user has to first register nodes:

1 **To register:**

POSTMAN:

1 Select POSTmethod

2 Enter the message in “body tab by selectingjson”.

3 Message: { “nodes”: [“http://**IP_Address(node2):Port_Number**”]}

4 Enter http://**IP ADDRESS (node1): PORT_NUMBER**/nodes/register

Terminal:

curl -X POST -H “Content-Type: application/json” -d ‘{ “nodes”: [“http://**IP_Address(node2):Port_Number**”] }’ “http://**IP_Address(node1):Port_No**”/transactions/new

2 To view commonchain:

Browser: http://**IP_Address(any node):Port_No**/nodes/resolve

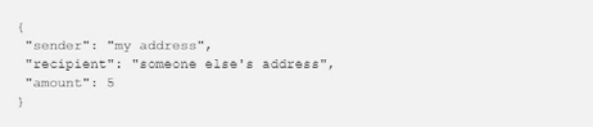
Terminal:

curl -X GET -H “Content-Type: application/json” “http://**IP_Address(Any-node):Port_No**/nodes/resolve”

Note: We used Raspberry pi model 3b for ARM processor and Wifi for communication between two devices.

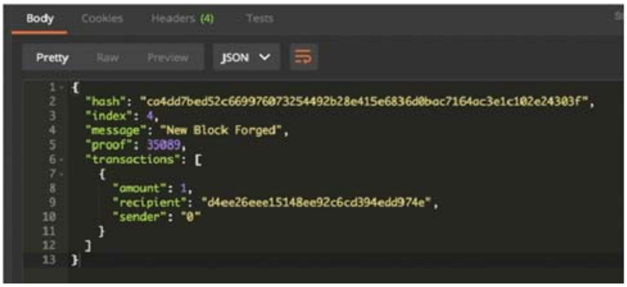
The block snippet is as shown in Figures 9–10.

Figure 9 Block content



```
{
 "sender": "my address",
 "recipient": "someone else's address",
 "amount": 5
}
```

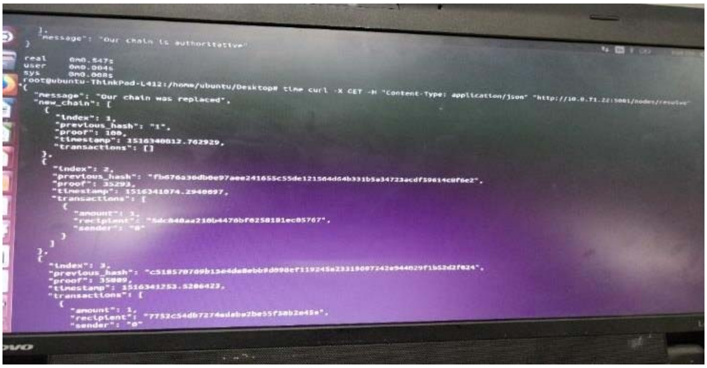
Figure 10 Detailed block content (see online version for colours)



```
{
  "hash": "ca4dd7bed52c669976073254492b28e415e6836d0bac7164ac3e1c102e24303f",
  "index": 4,
  "message": "New Block Forged",
  "proof": 35089,
  "transactions": [
    {
      "amount": 1,
      "recipient": "d4ee26eee15148ee92c6cd394edd974e",
      "sender": "0"
    }
  ]
}
```

The complete blockchain snippet:

Figure 11 Blockchain formed after message passing (see online version for colours)



```
    ],
    "message": "Our chain is authoritative"
}

real    0m0.547s
user    0m0.004s
sys     0m0.008s
root@ubuntu-ThinkPad-L412:/home/ubuntu/Desktop# time curl -X GET -H "Content-Type: application/json" "http://10.0.71.22:5001/nodes/resolve"
{
  "message": "Our chain was replaced",
  "new_chain": [
    {
      "index": 1,
      "previous_hash": "1",
      "proof": 100,
      "timestamp": 1516340812.762929,
      "transactions": []
    },
    {
      "index": 2,
      "previous_hash": "fb676a30db0e97aee241655c55de121564d64b331b5a34723acdf59614c8f6e2",
      "proof": 35293,
      "timestamp": 1516341074.2940897,
      "transactions": [
        {
          "amount": 1,
          "recipient": "5dc848aa210b4476bf0258181ec05767",
          "sender": "0"
        }
      ]
    },
    {
      "index": 3,
      "previous_hash": "c518578709b13e4da8ebb8d098ef119245a23319007242a944029f1b52d2f024",
      "proof": 35089,
      "timestamp": 1516341253.5206423,
      "transactions": [
        {
          "amount": 1,
          "recipient": "7752c54db7274adaba2be55f3ab2a45a",
          "sender": "0"
```

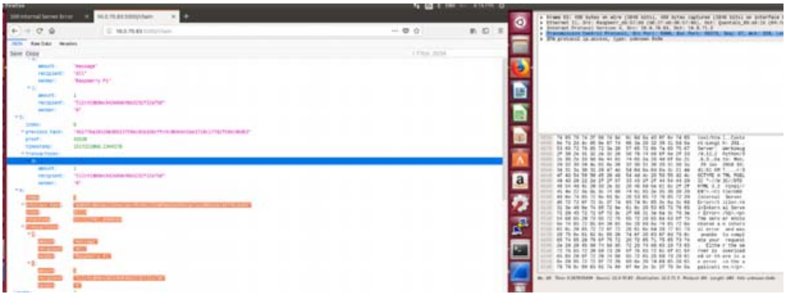
4.1 Security implications

For the addition of security in message transmission, we introduce Base64 encoding of the message so that it can't be decoded during network packet analysis it transmits in a decoded form.

During the mining phase, we perform message encoding using Base64 encoding. Due to this encoding, the transmitted message gets encrypted. Since Base64 encoding is itself a light weight encryption algorithm, hence it does not procure any additional processing power in computation

The wireshark analysis is encrypted in Figure 12.

Figure 12 Wireshark analysis of transmitted message (see online version for colours)



4.2 Distributed consensus

In Bitcoin, the distributed consensus is achieved by means of a proof-of-work (PoW) approach. To produce a valid block and add it to blockchain (mining process), a peer has to perform an extremely time-consuming work characterised by low success probability. More precisely, the miner has to randomly hash the block header until a value below a target threshold is obtained. To encourage the competition between miners, a reward is given to the first one that completes the work.

In the system we propose, the blockchain is built by means of a proof-of-stake (PoS) approach, whereby next block in the blockchain is the one produced by the peer that has obtained the majority of proofs of data in the latest T blocks of the blockchain. No time-consuming and energy-hungry work is required for mining valid blocks. Thus, block mining is not rewarded. If a peer receives more than one valid block from its neighbours, it will add to the end of its blockchain the block produced by the peer with the largest number of proofs of data, in the latest T blocks. The latest T blocks of the chain cannot include more than one block produced by the same peer. This is to prevent the monopoly problem, i.e., a peer that keeps out the proofs of data that concern other peers from the block it produces, in order to remain the owner of most proofs of data and, therefore, to take control of the blockchain.

4.3 Robustness analysis

In this section, we analyse the robustness of the proposed scheme with respect to all major smart home-related attacks.

4.3.1 Robustness against the DoS

Denial of service (DoS) attacks are the most common attacks in a smart home environment. DoS attacks exhaust service provider resources and network bandwidth. Since blockchain technology is based on peer to peer communication as there is no centralised server through which communication is happening. So it is resilient against the DoS attacks.

4.3.2 Robustness against the eavesdropping

Passive attackers can target various communication channels (e.g., wireless networks, local wired networks, Internet) to extract data from the information flow. Obviously, an internal attacker that gains access to an infrastructure will be able to extract the information that circulates within that infrastructure. But since we have applied encryption for our message it is impossible for an intruder to analyse the packet.

4.3.3 Robustness against the masquerading

In this type of attack, an attacker uses fake identity with the aim of gaining unauthorised access to the network through legitimate access identification. If the authorisation process is not fully protected then this attack can be extremely vulnerable to the network. Here in this attack, the main mindset of the intruder is to get the secret data or to acquire authentic services. But since device, before addition is verified by using device signature which will acts as a device private key since if the device is malicious it will generate the message with a random hash, which will not be added in the blockchain.

4.3.4 Robustness against the de-synchronisation attack

The main motive of de-synchronisation attack is to disturb the established connection between the two end-points or devices in a smart home network. Here in this attack, the attacker tries to modify the control flags as well as the sequence number in order to alter the packets, or messages that take place between the two devices or the end points. Since data in blockchain is connected through Hash of the previous block. Any alteration in Hash or data modification will lead to generating a wrong hash code, thus it will not be able to insert in the main chain. Thus it prevents it from de-synchronisation attacks.

5 Performance evaluation

For performance evaluation of our proposed scheme, we use Raspberry pi for communication as well as Wi-Fi for connection. Here we consider three-time factors:

1 transaction time (T_{time}): it denotes how much time it is taking for performing a transaction process for a single node (n)

2 mine time (M_{time}): it denotes how much time the miners will take to validate the block and add it to the block chain

3 chain time (C_{time}): it denotes the time taken for the chain formation for n devices.

Since transaction time for a device is constant it will always be less than the mine time

$$T_{time} < M_{time} \text{ such that } T_{time} > 0$$
$$C_{time} = T_{time} + M_{time}$$

The experimented scenario consists of a static network of eight peers having four devices each and each device having sensors which are communicating using Wi-Fi. The minimum fraction of consensus to be achieved in order to accept for the addition in the chain is 51%.

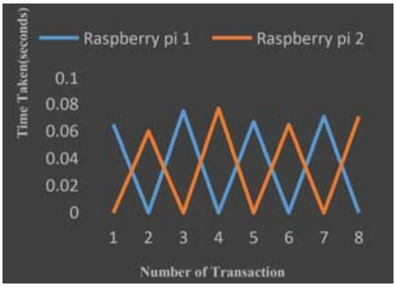
5.1 Transaction

Time taken in a transactions between Raspberry pi 1 and Raspberry pi 2.

Table 1 Transaction time comparison between two Raspberry pi's

Transactions	*Raspberry pi 1 (time taken)*	*Raspberry pi 2 (time taken)*
1	0.065	0
2	0	0.06
3	0.075	0
4	0	0.077
5	0.067	0
6	0	0.065
7	0.071	0
8	0	0.071

Figure 13 Transaction time comparison between two Raspberry pi's (see online version for colours)

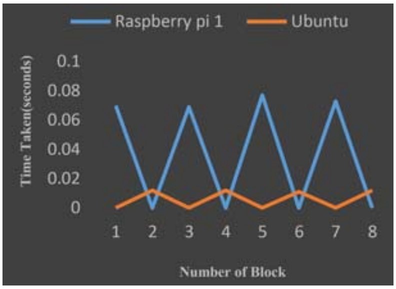


Time taken in a transaction between Raspberry pi 1 and Ubuntu.

Table 2 Transaction time comparison between Raspberry pi and Ubuntu

Transactions	*Raspberry pi (time taken)*	*Ubuntu (time taken)*
1	0.07	0
2	0	0.012
3	0.069	0
4	0	0.012
5	0.077	0
6	0	0.011
7	0.073	0
8	0	0.012

Figure 14 Transaction time comparison between Raspberry pi and Ubuntu (see online version for colours)



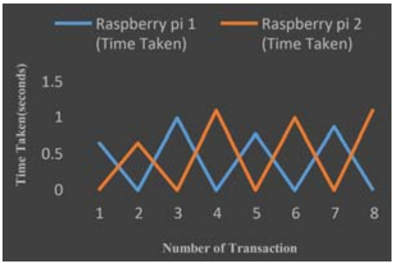
5.2 Mine

Time taken in mining between Raspberry pi 1 and Raspberry pi 2.

Table 3 Mine time comparison between two Raspberry pi's

Transactions	*Raspberry pi 1 (time taken)*	*Raspberry pi 2 (time taken)*
1	0.665	0
2	0	0.649
3	0.99	0
4	0	1.093
5	0.773	0
6	0	0.994
7	0.873	0
8	0	1.11

Figure 15 Time taken in mining between Raspberry pi 1 and Raspberry pi 2 (see online version for colours)

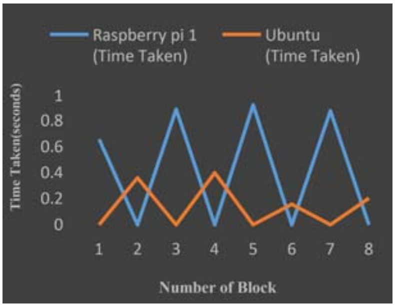


Time taken in mining between Raspberry pi 1 and Ubuntu.

Table 4 Mine time comparison between Raspberry pi and Ubuntu

Transactions	*Raspberry pi (time taken)*	*Ubuntu (time taken)*
1	0.66	0
2	0	0.36
3	0.89	0
4	0	0.4
5	0.923	0
6	0	0.157
7	0.877	0
8	0	0.203

Figure 16 Time taken in mining between Raspberry pi 1 and Ubuntu (see online version for colours)



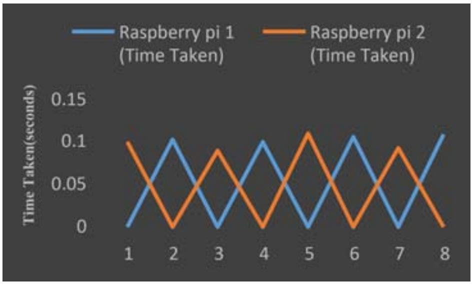
5.3 Chain

Time taken in chain between Raspberry pi 1 and Raspberry pi 2.

Table 5 Chain time comparison between two Raspberry pi's

Transactions	*Raspberry pi 1 (time taken)*	*Raspberry pi 2 (time taken)*
1	0	0.099
2	0.102	0
3	0	0.089
4	0.099	0
5	0	0.109
6	0.105	0
7	0	0.092
8	0.108	0

Figure 17 Time taken in forming chain between Raspberry pi 1 and Raspberry pi 2 (see online version for colours)

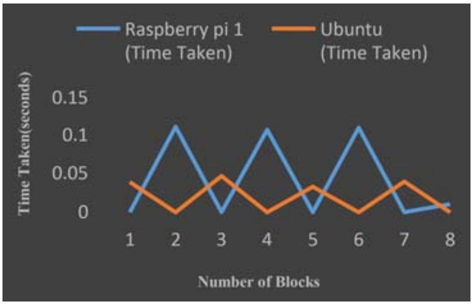


Time taken in chain between Raspberry pi 1 and Ubuntu.

Table 6 Chain time comparison between Raspberry pi and Ubuntu

Transactions	*Raspberry pi (time taken)*	*Ubuntu (time taken)*
1	0	0.039
2	0.111	0
3	0	0.047
4	0.107	0
5	0	0.033
6	0.11	0
7	0	0.04
8	0.0102	0

Figure 18 Time taken in forming chain between Raspberry pi 1 and Ubuntu (see online version for colours)

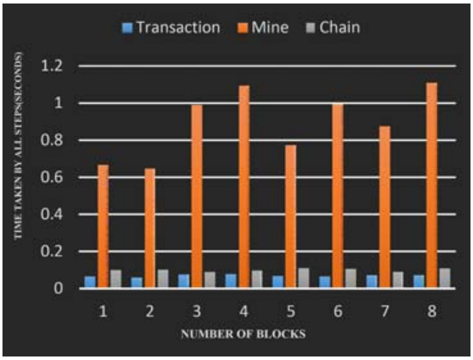


5.4 Blockchain

Table 7 Blockchain time taken overall for three processes

Transactions	*Transactions (time taken)*	*Mine (time taken)*	*Chain (time taken)*
1	0.065	0.665	0.099
2	0.06	0.649	0.102
3	0.075	0.99	0.089
4	0.077	1.093	0.099
5	0.067	0.773	0.109
6	0.065	0.994	0.105
7	0.071	0.092	0.092
8	0.071	1.11	0.108

Figure 19 Overall time taken by whole blockchain formation process (see online version for colours)



6 Conclusions

In this paper, we have presented a novel approach for producing proofs of data, i.e., digital certificates that attest someone's sent messages, at some point in time whereby SBSs can validate device data. We have illustrated a completely decentralised, blockchain-based peer-to-peer scheme that guarantees data trustworthiness and preserves user privacy, at the same time. We have analysed the robustness of the proposed scheme against all major Smart Home-related attacks. Furthermore, we have presented a preliminary simulation-based performance evaluation of the proposed scheme. Regarding future work, we plan to implement the proposed scheme in IotPot so that the scheme works more efficiently for malware infection and eliminate security risks against the other networks threats.

References

Chen, P-y. (2014) 'Information fusion to defend intentional attack in internet of things', *IEEE Internet of Things Journal*, IEEE, Vol. 1, No. 4, pp.337–348.

Christidis, K. and Devetsikiotis, M. (2016) 'Blockchains and smart contracts for the internet of things', *IEEE Access*, Vol. 4, pp.2292–2303, ISSN: 2169-3536.

Conti, M., Kumar, E.S., Lal, C. and Ruj, S. (2017) 'A survey on security and privacy issues of bitcoin', *IEEE Communications Surveys & Tutorials*, Vol. 20, No. 4, pp.3416–3452.

Dorri, A., Kanhare, S.S. and Jurdak, R. (2017a) *Blockchain in Internet of Things Challenges and Solutions*, December [online] https://arxiv.org/abs/1608.05187?context=cs.

Dorri, A., Kanhare, S.S., Jurdak, R. and Gauravaram, P. (2017b) *LSB: A Lightweight Scalable BlockChain for IoT Security and Privacy*, December [online] https://arxiv.org/abs/1712.02969?context=cs.

Dorri, A., Kanhere, S.S. and Jurdak, R. (2017c) 'Towards an optimized blockchain for IoT', *Proceedings of 2nd ACM/IEEE International Conference on Internet-of-Šings Design and Implementation*, Pitsburgh, PA, USA, April.

Henrik, J., Garcia, O. and Wehrle, K. (2015) 'Privacy in the internet of things: threats and challenges', *Security and Communication Networks*, Vol. 7, No. 12, pp.2728–2742.

Henstchel, K., Jacob, D., Singer, J. and Chalners, M. (2016) 'Supersensors: Raspberry pi devices for smart campus infrastructure', *IEEE 4thInternational Conference on future Internet of Things & Cloud (FiCloud)*, IEEE, pp.58–62.

https://brilliant.org/wiki/merkle-tree/ (accessed 2 February 2018).

https://www.tcs.com/content/dam/tcs/pdf/technologies/internet-of-things/abstract/Blockchain%20for%20the%20IoT.pdf (accessed 2 February 2018).

Kopetz, H. (2011) 'Internet of things', in *Real-Time Systems*, pp.307–323, Springer, New York, NY.

Mendez, D., Papapanagiotou, I. and Yang, B. (2017) 'Internet of things: survey on security', *Information Security Journal: A Global Perspective*, Vol. 27, No. 3, pp.162–182.

Stanciu, A. (2017) *21st International Conference on Control Systems and Computer Science*, IEEE.

Zaghal, A. (2016) *Wikipedia Merkle Tree* [online] https://en.wikipedia.org/wiki/Merkle_tree (accessed 27 April 2016).