Sliding Window Protocol for Internet of Things

Md. Aaqeel Hasan, Jaypal Medida, N. Laxmi Prasanna.

Abstract*: Internet of Things (IoT) refers to the concept of connecting non-traditional computers and related sources with the help of the internet. This includes incorporating basic computing and communication technologies for daily use into Physical things. Security and Confidentiality are two major challenges in IoT. In the current security mechanisms available for IoT, the limitations in the memory, energy resources, and CPU of IoT devices compromises the critical security specifications in IoT devices. Also, the centralized architectures for security are not appropriate for IoT because of a Single attack point. It is costly to defend against attacks targeted on centralized infrastructure. Therefore, it is important to decentralize the IoT security architecture to meet the requirements of resource constraints. Blockchain is a decentralized encryption system with a large number of uses. However, because of its high computational complexity and poor scalability, the Traditional Blockchain environment is not suitable for IoT applications. So, we introduce a Sliding window protocol to the traditional blockchain so that it will better suit the applications in the IoT environment. Changing the conventional blockchain and introducing a sliding window to it makes it use previous blocks in proof of work to shape the next hash block. SWBC's results are analyzed on a data stream generated from an IoT testbed (Smart Home) in real-time. The results show that the proposed sliding window protocol improves security and reduces memory overhead and consumes fewer resources for Security.

***Keywords*: Blockchain, Sliding Window, Merkel Tree, Window Size, Miners, Proof of Work.**

I. INTRODUCTION

Blockchain is a distributed ledger. What that means is that it is basically a distributed database where the source of truth can be verified at any point around the network. it is used to record transactions between two or more parties. And there is no centralized control over any of the transactions. In contrast to relational database systems where the data can be both appended and modified by the users, Blockchain attaches all the novel entries to the end of the ledger. This means that the user will not have permission to modify the data in the existing blockchain network. Also, all the other parties must verify using a consensus algorithm to add any new entries in the existing blockchain network. So, Having the feature of distributed ledger, attackers would face difficulties in stealing or manipulating the data as compared to the traditional database. As the traditional relational database serves as a centralized ledger and blockchain can have a decentralized record of transactions it can be used for many applications. There are many types of blockchain.

1. Public Blockchain (Permission-less).
2. Private Blockchain (Permissioned).
3. Consortium Blockchain (Semi- Decentralised).
4. Hybrid Blockchain.

A Public Blockchain is a non-restrictive also known as a Permission-less distributed ledger. Anyone who has access to the internet can become an authorized node and be a part of the public blockchain network. Some examples of Permission-less ledgers are Bitcoin, Ethereum, Litecoin, and many other major Cryptocurrency Transactions.

Private Blockchain is a restrictive also known as Permissioned network. It is a closed network that is generally used within an organization where only some selected members are the participants of the network. Basically, Permissioned networks are similar in use as permission-less networks with some restrictions. Some examples of Private blockchain are Multichain and Hyperledger Projects.

In a semi-decentralized network, more than one organization can act as a node the blockchain transactions. these are generally used by Governments and Banks.

Firms like Energy web foundation and R3 etc employ this type of Blockchain network.

The Hybrid Blockchain as the name denotes is a combination of both Permissioned and Permission-less networks. It uses the features of both Systems and with such a hybrid network, users can have control over who gets access to which data that is stored in the blockchain network. Dragon chain employs this type of networks

Any of the Conventional Blockchain network methods are not appropriate to use in the applications of IoT with real-time data streams because of the complexity in the computation. The more the computation time, the more it becomes hard to maintain security in blockchain for the applications of IoT.

There are two main problems in employing blockchain architecture in the environment of IoT.

1. Computational complexity
2. Scalability.

Manuscript received on August 28, 2021.
Revised Manuscript received on September 05, 2021.
Manuscript published on September 30, 2021.
* Correspondence Author

Md. Aaqeel Hasan*, Department of Computer Science, Mallareddy College of Engineering and Technology, MRCET Campus, Hyderabad, India. Email: mohommad.aaqeel@gmail.com

Dr. Jaypal Medida, Associate Professor, Department of Computer Science, Mallareddy College of Engineering and Technology, MRCET Campus, Hyderabad, India. Email: mrcetmtcsecoord@gmail.com

N. Laxmi Prasanna, Department of Computer Science, GITAM University, Vizag, Andhra Pradesh, India Email: prassunune@gmail.com

The Computational Complexity is proportional to the Merkel tree. As the Merkle tree size increases the Difficulty level increases and thus increasing the computational complexity of the network. The Merkel tree-labels each leaf node with the hash of transactional data and all non-leaf nodes with a cryptographic hash of their child nodes. The number of transactions done in the Merkle tree thereby increases the time needed to process the IoT network.

Scalability on the other hand refers to the number of transactions that the blockchain network will handle over a certain period.

Bitcoin and other cryptocurrencies are a great example of the illustration of the functioning of blockchain. Bitcoin is a decentralized payment system that does not depend on a central authority to control the supply of its funds. In Bitcoin Blockchain, the block size is restricted to 1MB, and every 10 Minutes a block is mined. Many of the existing studies suggest that blockchain can be implemented for the security and privacy algorithms in IoT applications using its distributed network.

Here we propose a new blockchain architecture for IoT applications, in particular with the automated smart home applications. A smart home monitors and gives reports on the state of the home. The in-home systems are automated and tracked using IoT-connected devices. The Smart home testbed is the smallest unit in the IoT community. The safety standardization of a smart home helps us in the designing of a smart automated city with the concepts. In a Smart home environment, the sensors produce real-time information data streams that allow tracking of the current state of the house. it also allows us in analysing the energy and also investigate any errors or incidents inside the home. The amount of data produced by the testbed is directly proportional to the number of sensors that are used and also the data acquisition frequency. Thus here, having proper data sampling is essential in generating useful information which can be then used and stored subsequently in the blockchain. the data volume contained in a blockchain specifies the overhead packets. In this regard, the blockchain architecture of our proposed sliding window protocol enhances security and reduces IoT overhead in a smart home environment.

II. MOTIVATION AND CONTRIBUTIONS

IoT is revolutionizing the computer world and is being used in many applications all around the globe, therefore security and privacy are necessary to be provided in the IoT environment because the data generated can be attacked being centralized database storage. But, the Limitations in the resources like the CPU, memory, and energy of the IoT environment make the algorithms for traditional centralized security inefficient. Therefore, introducing the Sliding window protocol with a decentralized security algorithm in the IoT applications improves Security.

The contributions for this research work are as follows:

1. Implementing an SWBC for IoT is proposed to provide security while considering the limitations of the IoT environment.
2. A testbed Simulating a smart home environment is set up to test and analyse the performance of SWBC architecture.
3. The result analyses the performance, and security of the proposed IoT architecture are carried out and monitored on the simulated testbed of Smart home.

The rest of the project is organized as following sections:

1. Explaining the work related to the blockchain approach briefly.
2. The preliminary section gives an introduction to blockchain.
3. problem statement.
4. proposed SWBC architecture
5. Experimental setup.
6. Performance Analysis.
7. Conclusion and Future scope.

Each section deals with a different issue on using the blockchain in IoT. While it has many uses using it, at the same time it also consists of wide range of disadvantages causing it to make many changes. So, each IoT application tries to change the blockchain network basing upon its own needs. Some use sliding window, while other use a distributed cloud, the main reason why using traditional blockchain in IoT is not encouraged because of the number of resources this takes to execute and also the time of execution.

We also set up an experimental house which includes various sensors and replicates the real time smart home, using the experimental setup we can use it to calculate the time taken and also the risks and advantages in the real time with a small-scale testbed.

III. LITERATURE SURVEY

Some many projects and journals give information about the use of blockchain in IoT, its advantages and disadvantages over the network, and about how they perform with limiting memory and also security and privacy of the network. Here, we will take a look at some of the recent publications that tried to use blockchain in IoT and also their advantages and disadvantages. “The beauty of Blockchain” Published by S. Kulkarni in 2016. It consists of making use of blockchain network and its distributed ledger in the use of IoT environment, by the use of distributed ledger technology, the records can be decentralized and could provide with more security. This journal consists of utilizing the traditional blockchain technique for IoT. But, using it is not so easy as it would consume more resources and it gives more latency. Here, having more latency is not advisable as IoT devices and IoT environment functions very fast with low latency, and by having an increase in time of execution it sets as a major drawback in the project. “Sliding window Blockchain for IoT” Published by Sarath babu and Prescilla. K, in 2019. In this, they tried to introduce a Sliding window so as to help to counter the drawback of latency issues in the use of Blockchain networks in the IoT environment. As we know that usage of traditional blockchain is not advisable in IoT they tried to implement blockchain by the sliding window protocol, which suggests that instead of sending one block of information every time we send an encrypted block, we send a sliding window that contains n number of blocks, where n can be any number but for an optimized smart home environment, they take n as 4.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

So, sending a block of data speeds up the processing time and also reduces the time to exchange data. Hence here we changed the traditional blockchain technique to better suit the IoT environment.

A. Sliding Window Blockchain by Prescilla

Here we propose a new blockchain architecture for IoT applications, in particular with the automated smart home applications. A smart home monitors and gives reports on the state of the home. The in-home systems are automated and tracked using IoT-connected devices. The Smart home testbed is the smallest unit in the IoT community. The safety standardization of a smart home helps us in the designing of a smart automated city with the concepts.

In a Smart home environment, the sensors produce real-time information data streams that allow tracking of the current state of the house. it also allows us in analysing the energy and also investigate any errors or incidents inside the home. The amount of data produced by the testbed is directly proportional to the number of sensors that are used and also the data acquisition frequency. Thus here, having proper data sampling is essential in generating useful information which can be then used and stored subsequently in the blockchain. the data volume contained in a blockchain specifies the overhead packets. In this regard, the blockchain architecture of our proposed sliding window protocol enhances security and reduces IoT overhead in a smart home environment.

“A Review on the usage of Blockchain”: Blockchain uses a distributed ledger that replicates a copy of the leader around the parties, thus avoiding the single failure point that is susceptible to exploitation. Yet blockchain is faced with crucial challenges In designing future housing models for health care, the design of intelligent homes plays an important role. Intelligent homes use IoT as their world of networking. IoT is an internet-based network of things embedded in sensors in order to gather and share data IoT is helping to link the physical and virtual resources of a smart home integrated with electronics, sensors, actuators, and software. Machomen is one of the first intelligent home projects to build an environmentally sound home. The officer aims to improve the comfort of the occupants and reduce running costs. In order to reach the objectives, the agent forecasts how people are mobile and how they use the system. Given the value of the smart home concept, adequate cyber-attack security for residential customers is important. In terms of processing and overhead storage, traditional protection solutions appear to be costly for IoT due to the limited computing power and memory of IoT devices. The resources are restricted existence of IoT devices that are involved in an intelligent home, therefore, makes traditional security solutions impossible. In consequence, clever homes are vulnerable to security. The main challenges in implementing traditional IoT protection mechanisms include (i) restriction of resources, (ii) heterogeneous communication protocols, (iii). Qu et al. described conventional policies for protection and privacy based on asymmetric encryption schemes as a result of their centralized key management system as difficult to enforce in an IoT setting. In this context, blockchain technology supports tracking, coordinating, conducting transactions, and collecting information from a large number of devices so that applications do not need a centralized cloud can be created. Blockchain is a decentralized network that makes transactions possible for all parties. In areas like finance, insurance business, and healthcare, the blockchain solution has been widely implemented. Khatri has shown that IoT security systems based on the blockchain can be strengthened significantly. Dori et al. suggested an architecture that uses a centralized immutable private ledger that runs inside a smart home at the local IoT network level to reduce the overhead and a decentralized public blockchain with superior devices for better confidence. Shen et al. used blockchain to secure the protection and privacy of the information on a smart home device. In order to promote the sharing of service resources in a cryptographically verifiable manner, Christakis et al. have used smart IoT contracts. The different blockchains offered for IoT applications are compared in Table I. Table I provides the degree of Proof-of-Work complexity that is relevant to IoT has not yet been experimentally analysed and literature has yet to investigate the viability of the Merkle tree for IoT. Our paper provides more insights into the above fields. n in an IoT setting: (i) The compute-intensive and time-consuming estimation of proof of work. Given the resource restriction of the majority of IoT systems and low latency in the majority of IoT applications, the conventional blockchain implementation becomes impossible. (ii) The implementation of the Merkle tree becomes a bottleneck for IoT because numerous sensors have been installed in standard IoT use. (iii) The protocols underlying blockchain establish substantial overhead network communication that cannot be used for IoT communication.

IV. PROBLEM STATEMENT

Blockchain is a distributed ledger that replicates a copy of a leader around parties. Thus, avoiding the single failure point that is susceptible to any forms of exploitation. However, for its use in an IoT setting, Blockchain faces few crucial challenges:

1. The work is intensive and time-consuming in computational terms. given the resource restriction in a majority of IoT Systems and low latency in the majority of IoT applications, the conventional Blockchain implementation becomes impossible.
2. The Implementation of the Merkle tree becomes a bottleneck for IoT because numerous sensors have been installed in standard IoT use.
3. The protocols underlying Blockchain establish substantial overhead network communications that cannot be used in IoT applications.

This paper aims to propose an architecture in a Blockchain to enhance IoT protection, reduce overhead memory and overhead networks. it also measures blockchain architecture performance in an automated smart home. The Prototype of an IoT environment has a heterogeneous network of various technologies and types of equipment like Wi-Fi, Bluetooth, Zigbee, etc. The complete environment of IoT in the smart home is connected to different organizations including the hospital, health insurance, the police, and the NGO. The Homeowner has the choice of connecting the desired organizations to the smart home network of IoT. In this prototype, the SWBC is used for securely storing the home state and transactions between organizations. The blockchain is privately owned and authorizes. the hash stored in the encryption key in the blockchain is only shared with the relevant entity. thus, the data produced by the smart home shall be encrypted. The clever contracts or the third-party organizations that are connected to the IoT network are only applied to the chain network if all the blockchain community members validate the bock jointly by the owner and organizations. This paper aims to propose an architecture in a Blockchain to enhance IoT protection, reduce overhead memory and overhead networks. it also measures blockchain architecture performance in an automated smart home. The Prototype of an IoT environment has a heterogeneous network of various technologies and types of equipment like Wi-Fi, Bluetooth, Zigbee, etc. The complete environment of IoT in the smart home is connected to different organizations including the hospital, health insurance, the police, and the NGO. The Homeowner has the choice of connecting the desired organizations to the smart home network of IoT. In this prototype, the SWBC is used for securely storing the home state and transactions between organizations. The blockchain is privately owned and authorizes. the hash stored in the encryption key in the blockchain is only shared with the relevant entity. thus, the data produced by the smart home shall be encrypted. The clever contracts or the third-party organizations that are connected to the IoT network are only applied to the chain network if all the blockchain community members validate the bock jointly by the owner and organizations.

V. PROPOSED ARCHITECTURE

Here we use Sliding window as a protocol that is being introduced in IoT using blockchain utilizes a window that slides across all the added blocks in the blockchain network with each addition of a new block. At starting, the window has only one block but as the blocks are added to the blockchain network the number begins to rise to n, with each increase in block size, window size also increases. When building a new block, the blocks that existed in the previous sliding window are used. And also, the block hash for blocks is generated by hashing blocks in a window of the proposed architecture of sliding window protocol. This protocol size specifies the number of previous blocks that the hash update function has used to execute. The blockchain sliding window has O(n) computer overhead for constant mine difficulty where n is the function. Then the protocol enhances the Blockchain record immutability. An incorrect miner includes previous blocks (n-1) and the size of the window is a block. Here the size of the window is hidden and is only shared with miners with the use of genesis block. The restriction of the chain is stored in system memory and the entire Blockchain network is saved to a private cloud. This includes the latest n blocks. The elder block is released from the window and also from IoT system memory when the block slides. Consequently, the overhead memory for storing blocks in IoT is decreased. The following section explores the structure of the SWBC and its relation to a Bitcoin blockchain.

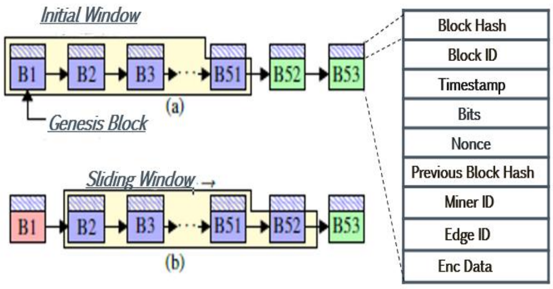


Fig: Architecture of a Sliding Window

This paper aims to propose an architecture in a Blockchain to enhance IoT protection, reduce overhead memory and overhead networks. it also measures blockchain architecture performance in an automated smart home. The Prototype of an IoT environment has a heterogeneous network of various technologies and types of equipment like Wi-Fi, Bluetooth, Zigbee, etc. The complete environment of IoT in the smart home is connected to different organizations including the hospital, health insurance, the police, and the NGO. The Homeowner has the choice of connecting the desired organizations to the smart home network of IoT. In this prototype, the SWBC is used for securely storing the home state and transactions between organizations. The blockchain is privately owned and authorizes. the hash stored in the encryption key in the blockchain is only shared with the relevant entity. thus, the data produced by the smart home shall be encrypted. The clever contracts or the third-party organizations that are connected to the IoT network are only applied to the chain network if all the blockchain community members validate the bock jointly by the owner and organizations.

VI. STRUCTURE OF SLIDING WINDOW IN BLOCKCHAIN

It displays the layout of the sliding window, Block Hash, Block ID, EdgeID, Nonce, MinerID, Timestamp and, bits. The Hash of a specific Block which is generated by hashing the current block and previous block(n-1). The Block ID is a single Block ID. The Block ID of the newly added block is permitted only for the members. The time field in the diagram shows the time of the creation of The specific Block. The area of Bits is the difficulty level of mining. The level of difficulty in mining is determined by the initial hash value and the number of zeroes. Four bits are represented for each zero. the levels of difficulty are as follows:

1. Tier 1 (4 Bit)
2. Tier 2 (8 Bit)
3. Tier 3 (12 Bit)
4. Tier 4 (16 Bit)
5. Tier 5(20 Bit)

As the number of zeroes rises, The Difficulty and complexity (Time of Calculation) in the level of mining increases. A High degree in the rise of complexity results in a high degree rise in the difficulty in Proof of Work which in turn causes it to consume much more computer resources. This is not ideal in applying Blockchain for the IoT environment. In addition, the degree of difficulty can be randomly selected from 1 to 5 to minimize the total calculation time to mine the blocks. MinerID represents the gateway identification and the Edge ID is the edge device identification.

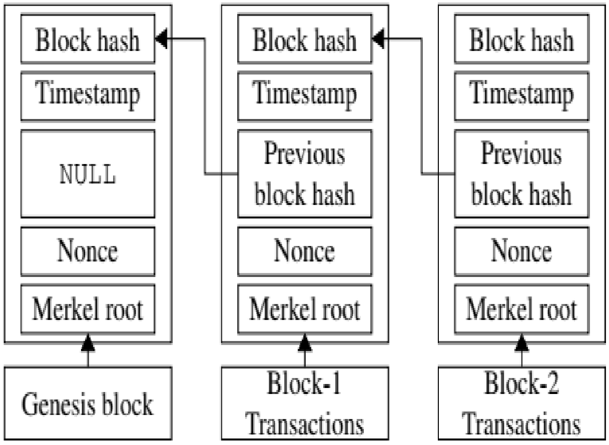


Fig: Blockchain Architecture

The hash value of the smart Contracts or the Third-party organizations that is agreed by all miners is the smart contract hash. Intelligent contract hash field is optional and this field activation ensures an intelligent re-entry contract. Our experiment does not include the Smart contract hash area. The EncData is made up of sensor data encrypted with a Password-based key derivation function which is also known as (PBKDF) and it follows Advanced Encryption Standard (AES) algorithm. In Traditional Blockchain parameters, the block size is restricted to 1MB in a Bitcoin Blockchain. But in a Sliding window architecture that is being introduced, with one block mining every 10 minutes, SWBC takes into account a variable block size with a 1MB maximum limit. The data will be separated into more than one block if the size of a block is greater than 1MB. Here we propose a new blockchain architecture for IoT applications, in particular with the automated smart home applications. A smart home monitors and gives reports on the state of the home. The in-home systems are automated and tracked using IoT-connected devices. The Smart home testbed is the smallest unit in the IoT community. The safety standardization of a smart home helps us in the designing of a smart automated city with the concepts.

In a Smart home environment, the sensors produce real-time information data streams that allow tracking of the current state of the house. it also allows us in analysing the energy and also investigate any errors or incidents inside the home. The amount of data produced by the testbed is directly proportional to the number of sensors that are used and also the data acquisition frequency. Thus here, having proper data sampling is essential in generating useful information which can be then used and stored subsequently in the blockchain. the data volume contained in a blockchain specifies the overhead packets. In this regard, the blockchain architecture of our proposed sliding window protocol enhances security and reduces IoT overhead in a smart home environment.

VII. EXPERIMENTAL SETUP

The experimental studies are defined using the following in this section:

(i) The use of Python and its communication protocols in intelligent home testbed modules and their features and (ii) the blockchain implementation. A. The prototype of a smart home in the sense of a smart home environment, a prototype IoT framework is implemented with SWBC. Figure 4 shows the IoT device testbed for the intelligent home. The prototype is comprised of sensors, lighting and fan electrical equipment, edge system (Arduino Uno), Wi-Fi (ESP8266), and gateway (personal computer). The environmental parameters are sensed using an ambient light sensor, a temperature sensor, a pressure sensor, a sensor for humidity, a fire sensor, and a sensor for hazardous gas.

The sensors sense a proximity Person who goes into and out of a room. The following functions apply to our intelligent home: I Relay 1 shall be closed when people are present in the room and ambient light during daytime shall be lower than the threshold value. (ii) Relax 2 is closed if people are in the room and the temperature reaches the threshold value. (iii) Alarms of the buzzer when fire and gas leakage occurs. (iv) When the sound is detected, the LED glows and people cannot sense theft in the house. (v) Read current value and transform the current sensor (ACS712) into the corresponding voltage between (0V to 5V). This sensor tests the voltage of the voltage from 0V to 1000VC (ZMPT101B). The energy consumed by the intelligent home is measured with the current sensor and sensor voltage values. (vi) The room status shall be marked time with Unix time from the NTP server. (vii) the sensor data is transported via the Wi-Fi module, ESP8266, via the TCP/IP protocol from the edge computer to the gateway. (viii) The sensed data on a PC is encrypted using the PBKDF2 Advanced Standard Encryption (AES) algorithm and safely saved with the Blockchain sliding window. The edge is an Arduino Uno with analog and digital pins to link the sensors. Relaxes is connected with 5V and Arduino's 3.3V pins have sensor connections. Linked sensor terminals are I digital/analog pins, (ii) Vcc (3.3V), and (iii) Arduino's GND. Arduino Uno's Tx and Rx are bound to ESP8266's Rx and Tx. The ESP8266 is designed to use AT commands to communicate with the TCP/IP protocol. ESP8266 is linked to a standard access point that transmits the server. The start, checking the ESP8266's readiness and creating the new TCP/IP link between the ESP8266 and the gateway take about 14 seconds for Arduino Uno. The above experiment is also performed as an edge computer on Arduino Due [32] and as a blockchain miner with Raspberry Pi.

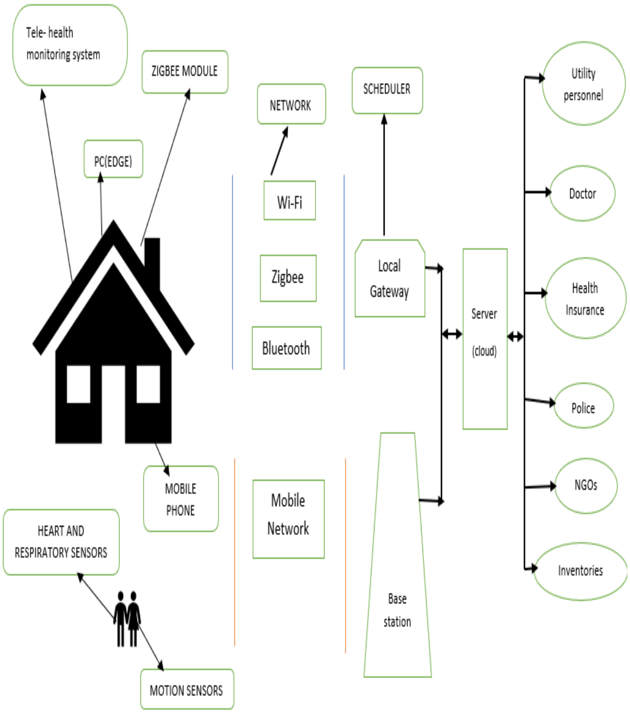


Fig: Experimental setup for SWBC Protocol

REALIZATION OF SLIDING WINDOW BLOCKCHAIN

The blockchain slide window will be introduced in the 4 GB memory processor Intel Core-i5-3470. It will run Ubuntu 16.04 LTS. Python is used to implement the blockchain algorithm. The smart homeowner chooses the miners according to his/her preferences for the blockchain. The owner knows the ID of each miner. Confidential correspondence. The owner and the miners shall have a session. The blockchain Genesis Builder is generated by the owner and transmitted to all miners with the necessary fields as specified in Figure 3. As the first block, miners add the genesis block. The block of genesis has the size of the window in the data sector. From the second block onwards, the intelligent home sensor data is registered. The community miners mined the block and returned the validation to the owner. The owner sends the consensus message to add the block when the confirmation is received by the miners. Both miners have the distinction of building a block and submitting it to the validation party (e.g., the biomedical data of a person is collected at the hospital). On the smart agreements established by the community are registered the access rights and privileges of the miners. Figure 5 illustrates the correspondence between IoT and the blockchain. The sliding window size increases from 1 to n as the blockchain builds up. Then the entire blockchain is saved in private cloud storage and the n-blocks window in the IoT devices. Sliding window reduces the overhead memory and makes blockchain on IoT devices possible. The smart device prototype parameters are shown in Table III. An average of 180 bytes of real-time data is produced from the intelligent home testbed. The data are encrypted over 100 bytes and over 160 bytes are blocked.

UML DIAGRAMS

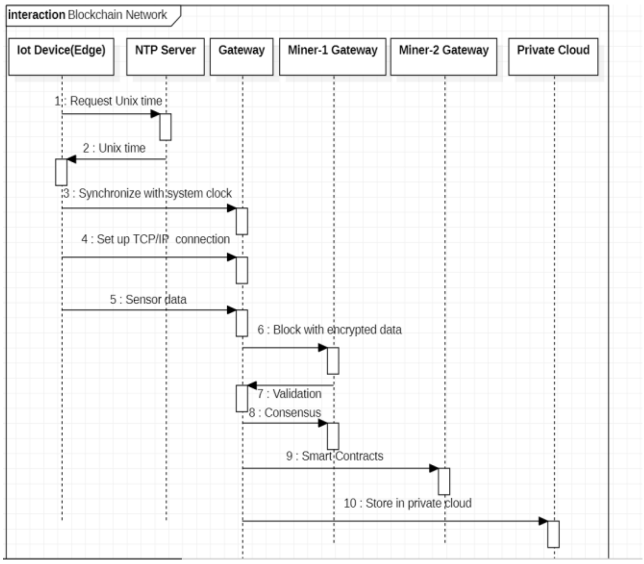


Fig: Communication between the Blockchain

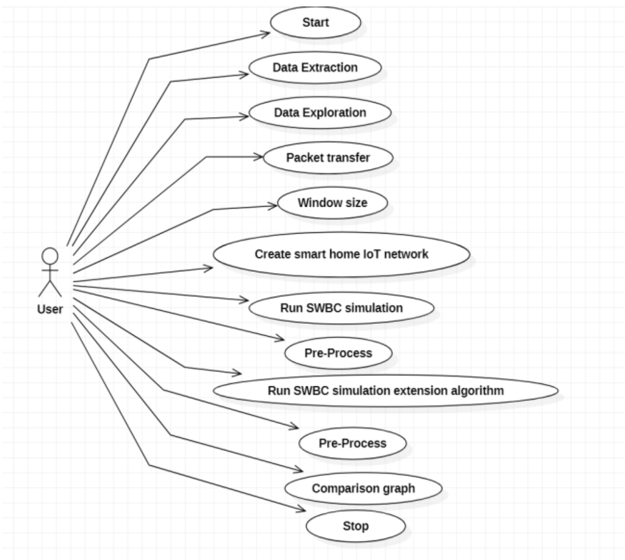


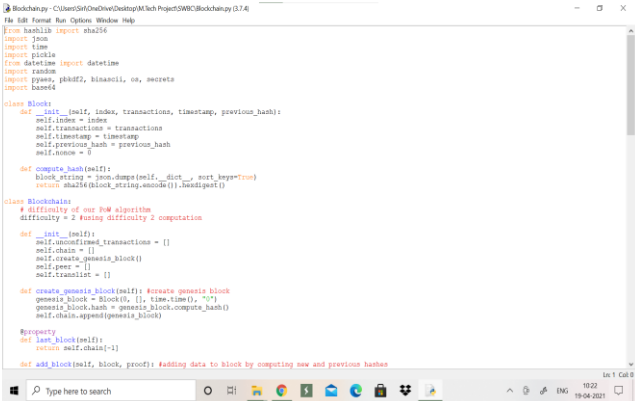
Fig: Use case Diagram

VIII. IMPLEMENTATION

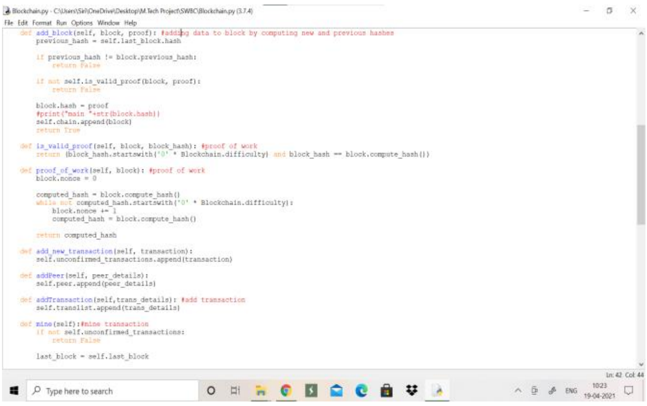
It describes the idea of IoT system protection with Blockchain technology since this technology supports decentralized data storage that means that data are stored in multiple nodes as compared with centralized storage where data is stored on one centralized server. Decentralized data storage offers data reception from any available node, with good reassurance when the Hash value of all nodes is checked by a single data store. Verification of all hash nodes is intensive and due to memory, CPU, and power consumption constraints, it cannot be extended to smaller IoT devices. To get over this problem author implements a sliding window technique, which fixes the window size, stores all Blockchain hash values and reduces the overhead of the old transaction blocks if the window size exceeds then, and maintains only new blocks, due to the technique of storage and data transfer.

In extension, the author says to continue saving energy I'm introducing the concept of time interval monitoring, and IoT will not process the data to be stored in Blockchain when sensors produce the same random data within a time interval and this duplicate avoidance will save additional energy.

This paper author uses sensors and other implementing devices, although we don't have any devices or sensors and I, therefore, simulate this project. Below the screen, the code shot shows how the hash chain is encrypted and blocked. Read the comment on the above screen to grasp the hash value and storage blockchain algorithm. Double-click on the 'run.bat' file to run the project under the screen. In the browser, pick the transfer number for a packet and then click on the 'Create Smart Home IOT Network' button to choose window sizes 5, 10, or 15. In the above screen, I have chosen the packet transfer number as 15, and the window size is 5 and the blockchain will store data up to 5 blocks. If the block is higher than that, the old block will then be removed and sent to the cloud for storage. In the above screen, every red-coloured circle is home IoT and the blue-coloured cloud is the full IoT-IoT info. Click on the button "Run SWBC Simulation" to make a random scan of the data in each circle while the label changes to red. The IOT13 label above detects data with a label colour shift to red colour and it will be used for 15 transfer packets and will be randomly selected for each transfer sensor. We can see in the text area which IOT senses data and which sensory meaning is divided by a comma-symbol. The next line shows encrypted AES data and then shows mining or not and is shown under the screen after simulation. In the above screen we get the above dialogue box after sending packets and sending total packets, and shows how long it took to process window 5 and showing total meaning and sending packets as 15. Underneath, we can view the new IoT storage blocks. In the screen above, our window size was 5 and the first block is blank for genesis, and 4 latest IOT logs view, and the first column shows the encrypted data in the top screen and the second column displays the decrypted data and the previous hash value. In the screen, above we can see the previous and existing hash value is checked by a blockchain. In the screen above we can see that the new first-line hash matches the previous second-line hash. In the screen above with the proposed work, we sense and store 15 packets and often the IoT sensor senses the same data that the temperature will not change for certain intervals. By tracking data, and by generating the same data again, we cannot process it in extension work. We can prevent this overhead. To prevent redundant processing, click on the button 'Run SWBC Simulation Extension Algorithm.' In the above screen, IOT also starts to detect and send packets, and on top of the screen, IOT10 is transformed into a red colour, which means its data is sensed and sent. Just 11 packets and 4 duplicate packets are prevented and this energy waste of four packets is saved in the top screen with extension work from 15 packages. Now click on the button 'Comparison Graph' In the diagram above, the X-axis represents algorithms names. Y-axis represents the number of packets transferred and only 11 packets can be saved with the extension application method.


Blockchain.py - C:\Users\Siri\OneDrive\Desktop\M.Tech Project\SWBC\Blockchain.py (3.7.4)
File Edit Format Run Options Window Help
Ln: 1 Col: 0
Type here to search
ENG
10:22
19-04-2021

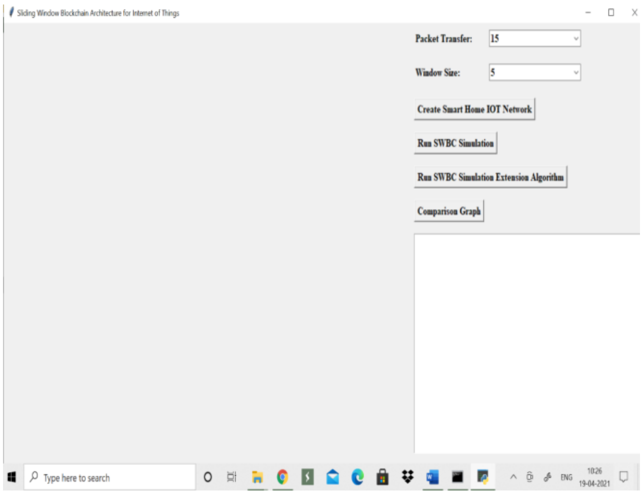
```
from hashlib import sha256
import json
import time
import pickle
from datetime import datetime
import random
import pyaes, pbkdf2, binascii, os, secrets
import base64

class Block:
    def __init__(self, index, transactions, timestamp, previous_hash):
        self.index = index
        self.transactions = transactions
        self.timestamp = timestamp
        self.previous_hash = previous_hash
        self.nonce = 0

    def compute_hash(self):
        block_string = json.dumps(self.__dict__, sort_keys=True)
        return sha256(block_string.encode()).hexdigest()

class Blockchain:
    # difficulty of our PoW algorithm
    difficulty = 2 #using difficulty 2 computation

    def __init__(self):
        self.unconfirmed_transactions = []
        self.chain = []
        self.create_genesis_block()
        self.peer = []
        self.translist = []

    def create_genesis_block(self): #create genesis block
        genesis_block = Block(0, [], time.time(), "0")
        genesis_block.hash = genesis_block.compute_hash()
        self.chain.append(genesis_block)

    @property
    def last_block(self):
        return self.chain[-1]

    def add_block(self, block, proof): #adding data to block by computing new and previous hashes
```


Blockchain.py - C:\Users\Siri\OneDrive\Desktop\M.Tech Project\SWBC\Blockchain.py (3.7.4)
File Edit Format Run Options Window Help
Ln: 42 Col: 44
Type here to search
ENG
10:23
19-04-2021

```
    def add_block(self, block, proof): #adding data to block by computing new and previous hashes
        previous_hash = self.last_block.hash

        if previous_hash != block.previous_hash:
            return False

        if not self.is_valid_proof(block, proof):
            return False

        block.hash = proof
        #print("main "+str(block.hash))
        self.chain.append(block)
        return True

    def is_valid_proof(self, block, block_hash): #proof of work
        return (block_hash.startswith('0' * Blockchain.difficulty) and block_hash == block.compute_hash())

    def proof_of_work(self, block): #proof of work
        block.nonce = 0

        computed_hash = block.compute_hash()
        while not computed_hash.startswith('0' * Blockchain.difficulty):
            block.nonce += 1
            computed_hash = block.compute_hash()

        return computed_hash

    def add_new_transaction(self, transaction):
        self.unconfirmed_transactions.append(transaction)

    def addPeer(self, peer_details):
        self.peer.append(peer_details)

    def addTransaction(self,trans_details): #add transaction
        self.translist.append(trans_details)

    def mine(self):#mine transaction
        if not self.unconfirmed_transactions:
            return False

        last_block = self.last_block
```

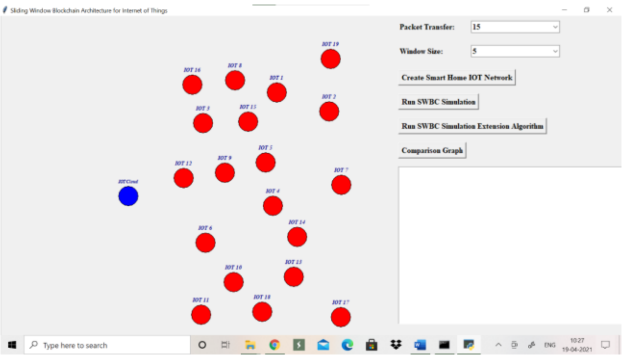
IX. TESTING AND RESULTS

Give the input for Packet transfer and then select window size as 5, 10 or 15 and then click on 'Create Smart Home IOT Network'


Sliding Window Blockchain Architecture for Internet of Things
Create Smart Home IOT Network
Run SWBC Simulation
Run SWBC Simulation Extension Algorithm
Comparison Graph
Type here to search
ENG
10:26
19-04-2021

Packet Transfer: 15
Window Size: 5

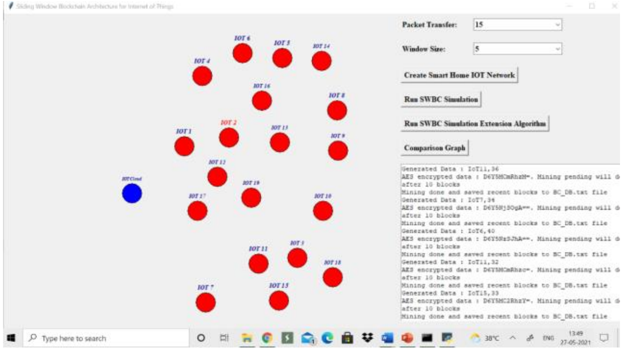
Packet transfer as 15 and window size is 5 and block chain can store data up to 5 blocks and if exceed then old block remove out and send to cloud for storage and new block will store in IoT memory. In above screen all red circles are home IoT and blue colour circle is the IoT cloud which will receive data from IoT upon IoT window full. Now click on 'Run SWBC Simulation' button to allow each circle to sense data randomly and while sensing circle label will change to red colour.



Packet Transfer: 15

Window Size: 5

IOT 2 label is sensing data and its label colour change to red colour and this simulation will run for 15 packets transfer and for each transfer sensor will be chosen randomly. In text area we can see which IoT is sensing data and its sense value separated by comma symbol. In next line displaying AES encrypted data and then displaying mining

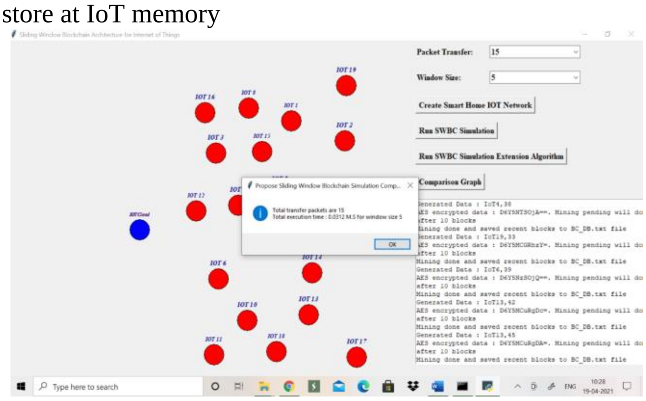


Packet Transfer: 15

Window Size: 5

```
Generated Data : IoT11,36
AES encrypted data : D6Y5MCmRhzM=. Mining pending will d
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT7,34
AES encrypted data : D6Y5Mj5OgA==. Mining pending will d
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT6,40
AES encrypted data : D6Y5Nz5JhA==. Mining pending will d
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT11,32
AES encrypted data : D6Y5MCmRhzc=. Mining pending will d
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT15,33
AES encrypted data : D6Y5MC2RhzY=. Mining pending will d
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
```

Sending packets, we will get above dialog box with total packets sense and it will display how much it took to process that window size 5 and displaying total sense and send packets as 15. In below screen we can see latest recent block store at IoT memory



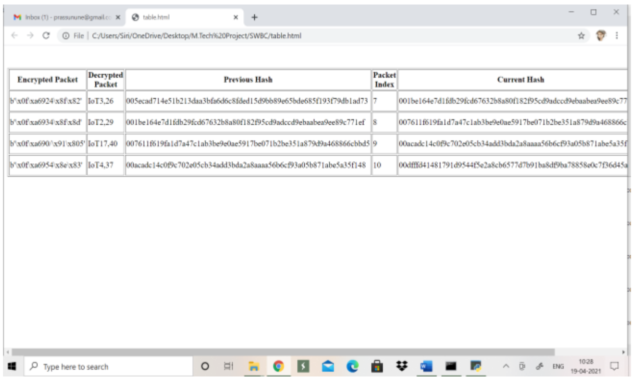
Packet Transfer: 15

Window Size: 5

Propose Sliding Window Blockchain Simulation Comp...

Total transfer packets are 15
Total execution time : 0.0312 M.S for window size 5

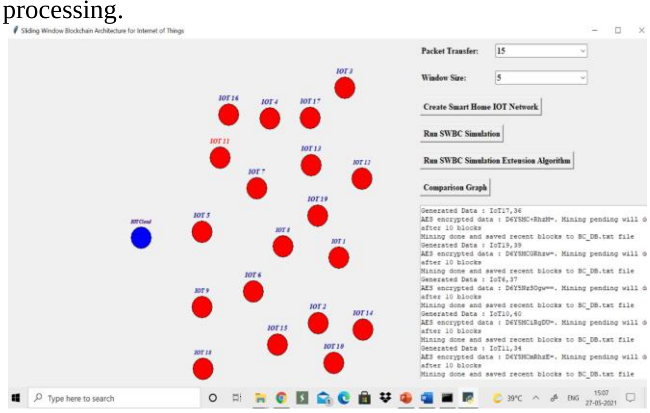
```
enerated Data : IoT4,38
ES encrypted data : D6Y5NTSOjA==. Mining pending will do
fter 10 blocks
ining done and saved recent blocks to BC_DB.txt file
enerated Data : IoT19,33
ES encrypted data : D6Y5MCGRhzY=. Mining pending will do
fter 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT6,39
AES encrypted data : D6Y5Nz5OjQ==. Mining pending will do
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT13,42
AES encrypted data : D6Y5MCuRgDc=. Mining pending will do
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT13,45
AES encrypted data : D6Y5MCuRgDA=. Mining pending will do
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
```



C:/Users/Siri/OneDrive/Desktop/M.Tech%20Project/SWBC/table.html

Encrypted Packet	Decrypted Packet	Previous Hash	Packet Index	Current Hash
b'\x0f\xa6924\x8f\x82'	IoT3,26	005ecad714e51b213daa3bfa6d6c8fded15d9bb89e65bde685f193f79db1ad73	7	001be164e7d1fdb29fcd67632b8a80f182f95cd9adccd9ebaabea9ee89c77
b'\x0f\xa6934\x8f\x8d'	IoT2,29	001be164e7d1fdb29fcd67632b8a80f182f95cd9adccd9ebaabea9ee89c771ef	8	007611f619fa1d7a47c1ab3be9e0ae5917be071b2be351a879d9a468866c
b'\x0f\xa690/\x91\x805'	IoT17,40	007611f619fa1d7a47c1ab3be9e0ae5917be071b2be351a879d9a468866cbbd5	9	00acadc14c0f9c702e05cb34add3bda2a8aaaa56b6cf93a05b871abe5a35f
b'\x0f\xa6954\x8e\x83'	IoT4,37	00acadc14c0f9c702e05cb34add3bda2a8aaaa56b6cf93a05b871abe5a35f148	10	00dfffd41481791d9544f5e2a8cb6577d7b91ba8df9ba78858e0c7f36d45a

Window size was 5 and first block is empty for genesis and latest 4 records of IoT are displaying and in the screen in first column showing previous hash value and then block chain index value and then current hash value with time. In above screen we can see that block chain verify previous and current hash value of first row is matched with previous hash of second row. The propose we can sense and store 15 packets and sometime IoT sensor will sense same data as temperature will not change for some intervals and if we can send same data again and again then it wastes processing time and increase overhead. We can avoid this overhead by monitoring data and if same data generate again then we will not process in extension work. Now click on 'Run SWBC Simulation Extension Algorithm' to avoid duplicate processing.

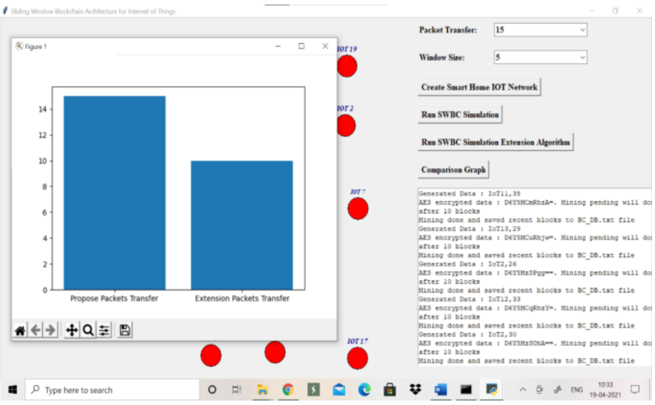


Packet Transfer: 15

Window Size: 5

```
Generated Data : IoT17,36
AES encrypted data : D6Y5MC+RhzM=. Mining pending will d
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT19,39
AES encrypted data : D6Y5MCGRhzw=. Mining pending will d
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT6,37
AES encrypted data : D6Y5Nz5Ogw==. Mining pending will d
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT10,40
AES encrypted data : D6Y5MCiRgDU=. Mining pending will d
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT11,34
AES encrypted data : D6Y5MCmRhzE=. Mining pending will d
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
```

IOT start sensing and sending packets and in above screen IoT 11 is change to red colour which means its sensing and sending data and after all 15 packets transfer will get below screen


Sliding Window Blockchain Architecture for Internet of Things
Figure 1
14
12
10
8
6
4
2
0
Propose Packets Transfer
Extension Packets Transfer
IOT 19
IOT 2
IOT 7
IOT 17
Create Smart Home IOT Network
Run SWBC Simulation
Run SWBC Simulation Extension Algorithm
Comparison Graph
Type here to search

Packet Transfer: 15

Window Size: 5

```
Generated Data : IoT11,35
AES encrypted data : D6Y5MCmRhzA=. Mining pending will do
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT13,29
AES encrypted data : D6Y5MCuRhjw=. Mining pending will do
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT2,26
AES encrypted data : D6Y5MzSPgg==. Mining pending will do
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT12,33
AES encrypted data : D6Y5MCqRhzY=. Mining pending will do
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
Generated Data : IoT2,30
AES encrypted data : D6Y5MzSOhA==. Mining pending will do
after 10 blocks
Mining done and saved recent blocks to BC_DB.txt file
```

Here Graph x-axis represents algorithm name and y-axis represents number of packets transfer and with extension work application process only 11 packets can save energy of 4 packets

X. CONCLUSION

Resources such as computer power, energy sources, and memory are limited by IoT devices. The standard protection algorithms for IoT are therefore not feasible. We introduced a sliding window blockchain that meets the requirements of an IoT network with restricted resources by decreasing the overhead memory and restricting the overhead of the computation.

The overhead memory is minimized by only keeping a small part of the blockchain in the private cloud, as specified by the IoT's sliding Fenster size and the entire blockchain. Computer overhead is reduced by the complexity level of 1 to 5 and the removal of the Merkle tree. The protection is improved by using the properties of n blocks of the sliding window to produce the block hash. Unable the previous (n−1) blocks and data on the window size are obtained, a false miner can mine a block. We have found the following from the experimental results:

(i) PoW calculation time increases exponentially for each difficulty level. (ii) By increasing the number of miners in the group, the total block addition time increases.

(iii) The hash calculation time increases linearly as the window size increases.

(iv) A random complexity selection for each block in a blockchain decreases the overall addition time of the block.

The effect of variable sliding windows can be analysed in future work. The IoT environment can be adapted to new consensus algorithms. Also, the Blockchain's power consumption can be analysed to learn more about the energy resources needed by an IoT system.

REFERENCES

1. "Sliding window Blockchain architecture for Internet of Things", Prescilla K, Associate Member, IEEE, Sarath Babu, Graduate Student Member, IEEE, and B. S. Manoj, Senior Member, IEEE. (Open source vol.6, June 2019).
2. "A review on the use of blockchain for IoT Things", T.M.F Carames and P.F Lamas(IEEE access,32 979-33 001, May 2018).
3. "Blockchain in the IoT environment", A.Dorri(August 2016).
4. "Challenges and solutions of blockchain", S.S Kanhere, R. Jurdak(arXiv preprint:1608.05187).
5. "Smarthome for building", IoT agenda online at https:// internetofthingsagenda.com.
6. "Smart Home Research", L. Jiang, D.Y Liu (International conference of ML and cybernetics August 2004).
7. "Towards the definition of IoT", available on theinstitute.ieee.org(May 2015).
8. "The Internet of things(IoT) for ambient assisted living", J. Wan, X. Gu, L. Chen (International conference for cyber-enabled distributed computing) Oct 2017.
9. "Novel anonymous key establishment protocol for isolated smart homes", D. Abbas-in-ezhad-mood(IEEE transactions April 2020).
10. "The role of prediction algorithms in the mavhome smart architecture". S.K Das, D.J Cook, A. Bhattacharya (IEEE wireless communications, Dec 2002).
11. "Blockchain-based credibility verification" C.Qu, M. Tao, J. Zhang(Security and communication networks).
12. "Securing smart home", C. Lee, L. Zappaterra(Conference on communications IEEE Aug 2018).
13. "Security challenges and security requirements of IoT", K. Choi, H.A Lee Choi (IEEE conference of Network security October 2014).
14. "Blockchain technology in finance", P. Treleven, R.G Brown(vol.50, no.9 pp.14-17 September 2017).
15. "To blockchain or not to blockchain: that is the question", V. Gatteschi, F. Lamberti (IT Professional, March 2018).
16. "Introducing the IoT department", P.A Laplante, B. Bamba (IT professional, vol.20, no.1 January 2018).
17. "Blockchain for healthcare and cloud-based security", C Esposito, A.D Santis(IEEE cloud computing January 2018).
18. "Can blockchain strengthen IoT environment", N.Kshetri(IT Professional, bol.19, no.4 August 2017).
19. "Towards an optimized blockchain for the Internet of things", A. Dorri, S.S Kanhere(In proceedings of the second international conference on IoT design. ACM August 2017).
20. "Secure data uploading scheme for a smart home system", J. Shen, C. Wang(Information Sciences, Vol.453, pp.186-197 July 2018).
21. "Blockchains and smart contracts for the internet of things", K. Christidis, M. Devetsikiotis(IEEE Access January 2016).
22. "Enigma: Decentralized computation platform with guaranteed privacy", G. Zyskind, O. Nathan(arXiv:1506.030471, 2015).
23. "Decentralized privacy-preserving healthcare", A.D Trivedi, P. Srivastava(International conference for IoT, September 2015).
24. "Smart Home System securing and data uploading", X. Huang, Z.H. Zhan(Vol.453, pp.186-197, July 2018).
25. "IoT agenda for building a smart home", (Online available at https://internetofthingsagenda.techtarget.com/definition/smart-home)

AUTHORS PROFILE

Md. Aaqeel Hasan is an M.tech Scholar from Mallareddy College, MRCET Campus, Dulapally, Hyderabad. He has his Masters degree in Computer Sciences. His area of interest lies in Networking and Internet of Things and has been a part of Contribution in this field right from his B.Tech degree. His area of specialization lies in Linux and Networking and has been an active member of Open-Source Community. He was also a part of Cloud based computing program organized by BSNL in the Vizag. Many of his networking and Social engineering projects could be found on his Github Page. Email: mohommad.aaqeel@gmail.com

Dr. Jaypal Medida is an Associate Professor in the Department of Computer Science Engineering at Mallareddy College MRCET campus Hyderabad. He has his doctorate in Computer Sciences and has been a contribution to various researches and publication. His researches include Machine Learning and deep learning. He has been an Associate professor in MRCET campus and has helped many Graduates and Post Graduates to learn and is a Guidance to chase their dreams. He has been a contribution to many research papers. Email: mrcetcsemtcoord@gmail.com

N. Laxmi Prasanna is an M.Tech Scholar from GITAM University Visakhapatnam. She has her Masters degree in Computer Science. Her area of interests lies in Cryptography and Bigdata. She has been a part of various Server Management Programs throughout her education. She also did special courses on Advanced Java and Cryptography to help in her Career. She also conducted many seminars on Cryptography and Blockchain in her M.tech and has been an active member of GUSAC Science Club In GITAM University. Email: prassunune@gmail.com