Blockchain in Smart Power Grid Infrastructure

Mohit Vashistha
Indian Institute of Information Technology Guwahati
Guwahati, Assam, India
mohit6b@gmail.com

Ferdous A Barbhuiya
Indian Institute of Information Technology Guwahati
Guwahati, Assam, India
ferdous@iiitg.ac.in

ABSTRACT

Internet of Things(IoT) is playing a vital role in making the digital world smarter with widely adopted applications. One of the important application of such a system is Smart Power Grid. A critical system like Smart Power Grid requires stringent security measures like immutability of the data sensed by IoT based sensors, verifiability of records etc. It is also important that the system works in a decentralised manner as there are multiple sources of power in a Grid which are from different administrative domain. This paper represents a system which will help to address concerns about security, privacy, Identity along with immutability and verifiability of records. The proposed system allows exchange and collection of data in a decentralized, peer-to-peer, trustless network of devices using BlockChain.

CCS CONCEPTS

• **Security and privacy → Trust frameworks**; • **Applied computing**;

KEYWORDS

Blockchain in Smart Grid, Smart Grid Security, Industrial IoT

ACM Reference Format:
Mohit Vashistha and Ferdous A Barbhuiya. 2019. Blockchain in Smart Power Grid Infrastructure. In *2019 ACM International Symposium on Blockchain and Secure Critical Infrastructure (BSCI '19), July 8, 2019, Auckland, New Zealand.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3327960.3332388

1 INTRODUCTION

Internet of Things (IoT) comprises "Things" (or IoT devices) which have remote sensing and/or actuating capabilities, and can exchange data with other connected devices and applications (directly or indirectly). All the connected IoT devices can collect data and process the data either locally or send the data to centralized servers or cloud-based application back-ends for processing [13]. A segment of the IoT, known as Industrial IoT or IIoT has drawn considerable attention in recent years. In IIoT, the concept of smart grid has found support from governments worldwide to address sing energy independence, climate change and dealing with emergencies etc.

In IIoT, problem arises while storing and processing data in secure manner. There are several security and trust issues associated with such data. To address these issues, blockchain can be used because of the underlying properties of Blockchain. Because of the power of these properties of blockchain our system enables peers in a decentralised, trust-less, peer-to-peer network to interact with each other without the need for a trusted intermediary [14].

Smart Grid posses very high amount of critical data due to the presence of various participants taking part in transmission, storage and various other power data processing. It collects the critical sensitive data of the users connected to the grid and the data of the specifications of all the devices connected to it. It also involves large amount of data transmission which is very critical and vulnerable in the centralised environment making it less secure. Also, the requirement of third party in centralised smart grid reduces the transparency in the whole grid network.

Blockchain will not only provide the decentralization to the whole system to remove the third party redundancy from the whole environment, but it will also enhance the security in the network by the use of crypto-graphic hash chaining and crypto-graphic digital signature technology. As all the processed data is very critical in nature, it needs a high amount of security to build trust among all the peers present in the network. Also, all devices which are connected also posses vulnerable data to be hacked. Blockchain's immutable nature will remove these vulnerabilities from the system. Blockchain will also secure the identity of the participants due to the unique cryptograpic signature for the participants and the transactions in the form of hashes.

In this paper, we proposed an architecture which provides secure data processing and storage with the use of blockchain technology. This technology provides a decentralized, peer-to-peer, trust-less and secure environment to the whole processed and the stored data. Blockchain will store the data in the form of hashes which are being generated by the various cryptographic algorithms to secure the data. The whole data will be processed in the form of transactions which will produce a unique transaction hash for all types of different types of transactions. This system will also provide the decentralised environment to all the peers present in the network which will remove the dependency of the system on the centralised systems/servers. It also automates the process flow with

BSCI '19, July 8, 2019, Auckland, New Zealand

ACM ISBN 978-1-4503-6786-8/19/07…$15.00
https://doi.org/10.1145/3327960.3332388

the help of the mining process which is happening as a background process for all types of transactions to verify the transactions by running a consensus algorithm to check whether a transaction is liable to enter in the blockchain or not.

Rest of the paper is organised as follows. Section II presents the related work in the area. Section III describes how BlockChain support IoT due to it's properties. Section IV provides the description of all the BlockChain related terminologies to under the basic terms/concepts used in BlockChain. Section V describes the application and architecture of the distributed application for Smart Power Grid. It also explains all types of events and functionalities, event attributes and the overall architecture and the event flow involved in the application. Section VI describes the whole process flow which is taking place in the application development. Section VII shows the performance and the results for the considered Ethereum BlockChain. Section VIII concludes the proposed BlockChain based Solution for Smart Grid.

2 RELATED WORK

The Bitcoin transactions are recorded in a public ledger called the Blockchain[2]. The Blockchain technology was introduced along with Bitcoin by Satoshi Nakamoto. IBM and Samsung have announced a collaboration to build decentralized IoT solutions by leveraging the Blockchain technology.[1] Trans Active Grid has developed a combination of software and hardware technologies that enable users to buy and sell solar energy from each other securely and automatically, using smart contracts and the Blockchain [12]. Filament has built an open technology stack based on Blockchain technology, to enable devices to discover, communicate, and interact with each other in a fully autonomous and distributed manner [7].

Slock is a smart lock technology called which enables real-world physical objects to be controlled by the Blockchain [10]. For example, owners of a Slock who wants to rent their real-world physical objects (such as houses, cars or bikes) can set a deposit amount and a price for using the objects. Users can find the Slocks using the mobile app and then make a payment in Ethers to rent the objects. After the transactions are validated on the Ethereum Blockchain network, the users get permission to open or close the Slocks with their smartphone. A smart contract is automatically enforced between the owner and the user. After the object is returned, the deposit minus the cost of the rental is returned to the user.

Despite a number of potential benefits, digital disruption poses many challenges related to information security and privacy. A security framework was designed that integrates the blockchain technology with smart devices to provide a secure communication platform in a smart city.[15]

3 BLOCKCHAIN PROPERTIES RELEVANT FOR IOT

- Integrity of ledger (Cryptographic hash function)
  A cryptographic hash function is a hash function which takes an input (or 'message') and returns a fixed-size alphanumeric string. The string is called the 'hash value', 'message digest', 'digital fingerprint', 'digest' or 'checksum'.[4]
- Authenticity of transactions (Elliptic Curve Digital Signature Algorithm)
  Elliptic Curve Digital Signature Algorithm or ECDSA is a cryptographic algorithm used by Bitcoin to ensure that funds can only be spent by their rightful owners.
  A few concepts related to ECDSA:
  1). private key, 2). Public key and 3). Signature [6]
- Privacy of transactions (Pseudonymity through crypto tools)
  A key aspect of privacy in blockchains lies in the use of private and public keys. Blockchain systems use asymmetric cryptography to secure transactions between users. [9]
- Identity of participants (Cryptographic signatures)
  A digital signature is a mathematical scheme for verifying the authenticity of digital messages or documents. A valid digital signature, where the prerequisites are satisfied, gives a recipient very strong reason to believe that the message was created by a known sender(authentication), and that the message was not altered in transit (integrity). [5]
- Auditability and Transparency (Cryptographic hash chain)
  A hash chain is the successive application of a cryptographic hash function to a piece of data. In computer security, a hash chain is a method to produce many one-time keys from a single key or password. For non-repudiation a hash function can be applied successively to additional pieces of data in order to record the chronology of the existence of data. [3]

4 TERMINOLOGY

- **BlockChain :** Blockchain is a distributed data structure comprising a chain of blocks. Blockchain acts as a distributed database or a global ledger which maintains records of all transactions on a Blockchain network. The transactions are time stamped and bundled into blocks where each block is identified by its cryptographic hash. The blocks form a linear sequence where each block references the hash of the previous block, forming a chain of blocks called the Blockchain. A Blockchain is maintained by a network of nodes and every node executes and records the same transactions. The Blockchain is replicated among the nodes in the Blockchain network. Any node in the network can read the transactions. [14]
- **Smart Contracts :** A smart contract is a piece of code that resides on a Blockchain and is identified by a unique address. A smart contract includes a set of executable functions and state variables. The functions are executed when

transactions are made to these functions. The transactions include input parameters which are required by the functions in the contract. Upon the execution of a function, the state variables in the contract change depending on the logic implemented in the function. Contracts can be written in various high-level languages (such as Solidity or Python) [11]. Language-specific compilers for smart contracts (such as Solidity or Serpent) are used to compile the contracts into byte code. Once compiled the contracts are uploaded to the Blockchain network which assigns unique addresses to the contracts. Any user on the Blockchain network can trigger the functions in the contract by sending transactions to the contract. The contract code is executed on each node participating in the network as part of the verification of new blocks. [14]

- **Ethereum :** Ethereum is an open and programmable Blockchain platform. Anyone can sign up for the platform and create an Ethereum account. Users can create and deploy smart contracts to the Ethereum platform and build decentralized applications. The platform is not owned or controlled by a single entity and is powered by the peers who run the Ethereum nodes.[14]
- **Ethereum Virtual Machine :** Ethereum Virtual Machine (EVM) is the run-time environment for smart contracts in Ethereum. The nodes in the Ethereum network run the EVM. The EVM runs as a sandbox and provides an isolated execution environment. All the nodes in the Blockchain network perform the same computations thus providing redundancy in the execution of smart contracts. While this massive amount of redundancy is not an efficient approach for execution, but it is required to maintain consensus in the network where there is no centralized authority or a trusted third-party. [14]
- **Accounts :** Ethereum has two types of accountsâĂŤExternally Owned Accounts (EOAs) and Contract Accounts. EOAs are the accounts which are owned and controlled by the users. Each EOA has an Ether balance associated with it. These accounts can send transactions to other EOAs or contract accounts. The contract accounts are controlled by the associated contract code which is stored with the account. The contract code execution is triggered by transactions sent by EOAs or messages sent by other contracts. [14]
- **Ether :** Ether is the currency which is used in the Ethereum Blockchain network. The miners in the Ethereum network receive mining rewards in the form of Ethers. The base unit of Ether is called Wei (where 1 Ether = 10 18 Wei). [14]
- **Gas :** Gas is the name of the crypto fuel which is consumed for performing the operations on a Blockchain network. All the transactions on the network are charged a certain amount of gas. While sending a transaction, the sender sets a gas price which represents the fee the sender is willing to pay for gas. The senders of the transactions are charged a gas fee, which is paid to the miners and the balance is refunded to the sender. The gas fee paid is proportional to the amount of work that is needed to execute the transaction, in terms of the number of atomic instructions. [14]
- **Transactions :** Transactions are the messages which are sent by Externally Owned Accounts (EOAs) to other EOAs or contract accounts. Each transaction includes the address of the recipient, transaction data payload and a transaction value. When a transaction is sent to an EOA, the transaction value is transferred to the recipient. When a transaction is sent to a contract account, the transaction data payload is used to provide input to the contract function to be executed. Transactions are signed by the sender's private key. Transactions are selected and included in the blocks in the mining process. The state of the network is changed only by the transactions which are selected for inclusion in the blocks. The transactions on a Blockchain network can be read by all the participant nodes in the network. [14]
- **Dapp :** A Decentralized Application (or Dapp) is an application that uses smart contracts. Dapps provide a user-friendly interface to smart contracts. A crypto currency application is an example of a Dapp that runs on a Blockchain network.[14]
- **Key Infrastructure :** Each Externally Owned Account (EOA) has a public-private key pair associated with it. The account address is derived from the public key. When a new EOA is created, a JSON key file is created which has the public and private keys associated with the account. The private key is encrypted with the password which is provided while creating the account. For sending transactions to other accounts, the private key and the account password are required.[14]

5 DISTRIBUTED APPLICATION FOR SMART POWER GRID WITH IOT

Smart Grid distributed application using blockchain is build to showcase the importance of blockchain in the security of the collected data and also for the processing of that data. As we are using blockchain for all the data which will be processed as transactions between machine-to-machine, peer-to-machine and peer-to-peer. Blockchain will act as a tamper proof and immutable storage for the data of all the transactions among several peers inside the system. The data present inside the network will be publicly accessible which is hashed and encrypted using various cryptographic techniques like public-private key, digital signature, etc. All the transactions are replicated over all the nodes in the network which will provide a heavy resistance to modify, update or delete the data over the blockchain. There will not be any requirement of third party intermediary as all the transactions will be verified using the process of mining(section IV(J)). It also prevents the peers from modifying, updating and deleting the transaction because of the immutable nature of the blockchain. This property will help to

build trust among the peers. Every transaction is associated with a time-stamp which will help the peers to track the transactions and the miners to verify the transaction.

5.1 Events and Functionalities

This Dapp mainly consists of 4 important types of events for the transactions inside a smart grid for the distribution of supply among the peers.

Storage Event - These events cover all the storage done by the producer inside the grid. These events will occur whenever storage type of event will happen inside the system. In these types of events the function in the smart contract will be invoked by creating an object of the deployed contract using the web3 library of javaScript language. Initially, the smart contract has been written in a high language called Solidity. This smart contract contains the function which will take the inputs for the storage event. This smart contract also includes the function which will return the number of inserted records. Firstly, user will compile the smart contract using the compilation script by using the wallet address of the web3 instance. Compilation will give us a json file which has the whole bytecode for the smart contract. Now, deployment script will use this bytecode to deploy the smart contract. As soon as the smart contract got deployed, user can invoke the storage event by calling the function after creating an object for the deployed smart contract. After this it will create a transaction to be mined. The mining process will take place in the background. As soon as the mining process finishes, the immutable record has been created into the blockchain. These storage can be sensed by the various types of sensors by detecting the capacity of the grid at various timestamps. As soon as storage event will occur, a alert and storage will be generated on the both side of the application, one is purchaser side and other is the generator/producer side. Ref. Algorithm 1

Purchase Event - These events covers all the purchasing done by the buyer after the creation of the storage event. This event can only occur when a storage event has already been occurred. These event includes all the details about the purchasing transaction done by the customer of the supply. In these types of events the function in the smart contract will be invoked by creating an object of the deployed contract using the web3 library of javaScript language. Initially, the smart contract has been written in a high language called Solidity. This smart contract contains the function which will take the inputs for the storage event. This smart contract also includes the function which will return the number of inserted records. Firstly, user will compile the smart contract using the compilation script by using the wallet address of the web3 instance. Compilation will give us a json file which has the whole bytecode for the smart contract. Now, deployment script will use this bytecode to deploy the smart contract. As soon as the smart contract gets deployed, user can invoke the purchase event by calling the function after creating an object for the deployed smart contract. After this it will create a transaction to be mined. The mining process will take place in the background. As soon as the mining process finishes, the immutable record has been created into the blockchain. Ref. Algorithm 2

Transfer Event - These events covers all the transfer of supply from the power grid to the customer after the request of supply from customer. An alert will be be produced as soon as we see a reduction from storage and an acknowledgement from the customer for this particular transfer. There will always be cost associated with each transfer which directly depends on the rate of the power supply and the amount of supply requested by the customer. In these types of events the function in the smart contract will be invoked by creating an object of the deployed contract using the web3 library of javaScript language. Initially, the smart contract has been written in a high language called Solidity. This smart contract contains the function which will take the inputs for the storage event. This smart contract also includes the function which will return the number of inserted records. Firstly, user will compile the smart contract using the compilation script by using the wallet address of the web3 instance. Compilation will give us a json file which has the whole bytecode for the smart contract. Now, deployment script will use this bytecode to deploy the smart contract. As soon as the smart contract gets deployed, user can invoke the transfer event by calling the function after creating an object for the deployed smart contract. After this it will create a transaction to be mined. The mining process will take place in the background. As soon as the mining process finishes, the immutable record has been created into the blockchain. Ref. Algorithm 3

Bill Creation Event - This events occurs when all the above three events has already been occurred. This event will have all the data captured by a sensor during storage, purchase and transfer event. All these inputs from sensors will make a file. Then a Bill creation event will occur in which the bill will be captured as a file. Then a request will generate to store that file in the inter-planetary file system(IPFS). As soon as this bill gets stored in IPFS, it will generate a unique document hash. Then a event will be generated to store this unique document hash in the blockchain under the name of the purchaser. Ref. Algorithm 4

5.2 Event Attributes Table

Storage	Purchase	Transfer
Grid Capacity	Purchase Quantity	Device Name
Available storage	Purchase Date	Transfer Amount
Grid Location	Associated Cost	Start Time
Rate		End Time

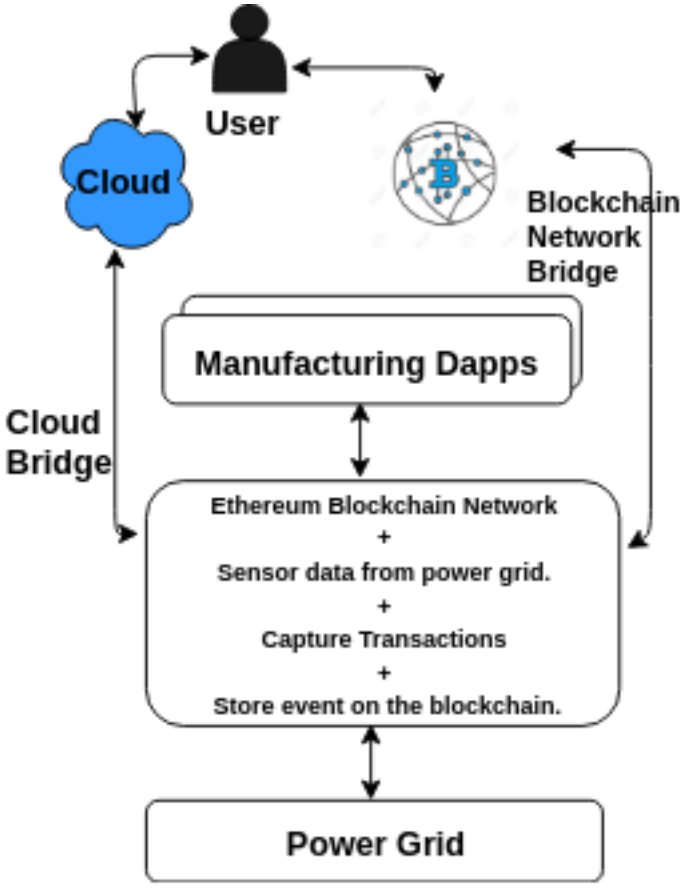

Figure 1: Overall Architecture.

Algorithm 1: Storage Event Algorithm

```
String gridCapacity, availableStorage, gridLocation, rate;
if ((gridCapacity) & (availableStorage) & (gridLocation) & (rate)
  are valid ) then
    sendRequest(gridCapacity, availableStorage, gridLocation,
      rate);
    publishToBlockChainRequest(gridCapacity,
      vailableStorage, gridLocation, rate) =>txnHash,
      timestamp;
    mining(txnHash, timestamp)
      if (mined properly) then
        publish the data to the blockchain;
        return blockNumber;
    else
        return "Error message - Could not mine";
    end
else
    return Input Error;;
end
```

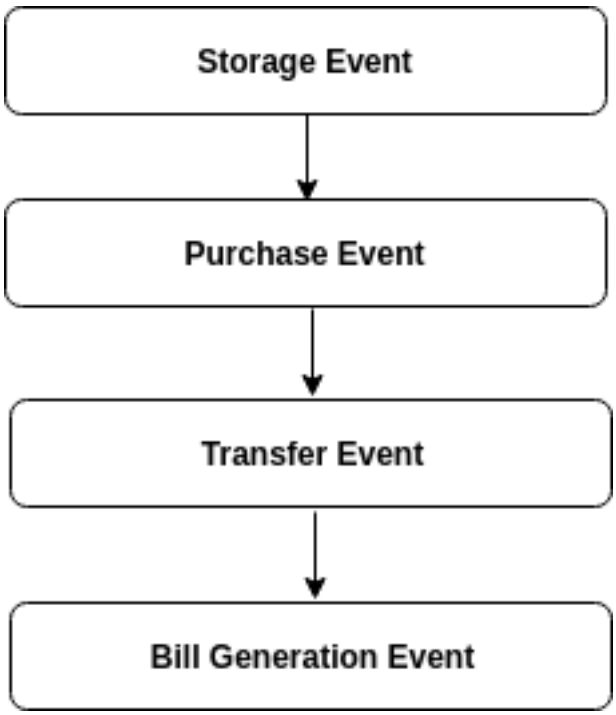

Figure 2: Event Flow.

Figure 3: Bill Event

5.3 Architecture

6 PROCESS FLOW

Different types of processes will be involved which will play a huge role in the enhancement over existing cloud based manufacturing models using blockchain technology by enabling consumer-to-machine and machine-to-machine transactions without a trusted intermediary, by automating machine maintenance and diagnostics tasks, by providing a distributed, secure and shared ledger of all transactions, assets and inventory records, all through the existence of a decentralized, trustless, peer-to-peer blockchain network. The key enabler component for the industrial machines in the proposed platform is the IoT device. The IoT device enables existing machines to communicate with the Blockchain network. The IoT device is a âĂIJplug and playâĂİ solution that allows machines to exchange data on their operations, send transactions to the associated smart

Algorithm 2: Purchase Event Algorithm

```
String purchaseQuantity;
if ( (purchaseQuantity is valid) & (purchaseQuantity <=
  availableStorage) ) then
    associatedCost = rate x purchaseQuantity;
    sendRequest(purchaseQuantity, associatedCost);
    publishToBlockChainRequest(purchaseQuantity,
      associatedCost) =>txnHash, timestamp;
    mining(txnHash, timestamp)
      if (mined properly) then
        publish the data to the blockchain;
        return blockNumber;
    else
        return "Error message - Could not mine";
    end
else
    return Input Error;;
end
```

Algorithm 3: Transfer Event Algorithm

```
String deviceName, transferAmount, startTime, endTime;
if ( (deviceName) & (transferAmount) & (startTime) &
  (endTime) are valid ) then
    totalCost = associatedCost;;
    sendRequest(deviceName, transferAmount, startTime,
      endTime, totalCost);
    publishToBlockChainRequest( deviceName,
      transferAmount, startTime, endTime, totalCost)
      =>txnHash, timestamp;
    mining(txnHash, timestamp)
      if (mined properly) then
        publish the data to the blockchain;
        return blockNumber;
    else
        return "Error message - Could not mine";
    end
else
    return Input Error;;
end
```

contracts and receive transactions from the peers on the Blockchain network.

Storage event will perform a machine-to-machine type of transaction in which IoT device will sense the capacity of the added storage and will pass the information to the blockchain network through smart contract as a blockchain event. As soon as the storage event gets added into the blockchain after the validation of

Algorithm 4: Bill Event Algorithm

```
file selectedFile;
if ( selectedFile is not currupt & empty ) then
    sendBillToIPFS(selectedFile) =>ipfsHash;
    publishipfsHashToBlockChain( ipfsHash) =>txnHash,
      timestamp;
    mining(txnHash, timestamp)
      if (mined properly) then
        publish the ipfsHash to the blockchain;
        return blockNumber;
    else
        return "Error message - Could not mine";
    end
else
    return Input Error;;
end
```

the transaction in the blockchain network, it will reflect inside the whole blockchain network. After the addition of storage event, purchase event will be of the consumer-to-machine type of transaction in which a event will be fired from the consumer end as soon as the IoT sensor sense the requirement on the consumer end to purchase the amount of power supply. An event will be generated to be stored into the blockchain through the underlying smart contract. As soon as this purchase event occurs, an alert will be generated which will be sensed by the IoT sensor and a machine-to-machine transaction will be generated in the form of transfer event which will send the requested amount of supply, IoT sensor will be activated throughout the supply period to sense any fault or the quantity of supply transferred to the user during the transfer period. Based on the supplied amount of quantity sensed by the IoT device, a bill event will get generated through the already written smart contract and and the cycle will get complete.

Underlying system will also provide the functionality of storing the document of the bill in the inter planetary file system which provides an unique document hash which can be stored in blockchain to give the ownership of the data attached with a bill to the user itself by removing the need of third party intermediary.

7 RESULTS

7.1 Performance Table

Gas Limit = 54246

Txn Type	GasPrice/Cost(ETH)	Txn/day
push/storage	3GWEI / 0.000163	60x24=1440
push/storage	4.5GWEI / 0.000244	1.67x60x24=2400
push/storage	20GWEI / 0.001085	2x60x24=2880
get/subscribe	0 / 0	15x60x24=21600

7.2 Experimental Setup and Performance Analysis

For a fixed gas limit of 54246 and gas price of 3GWEI, 4.5GWEI, 20GWEI, three type of slow rate, medium rate and fast rate transaction can happen with the transaction cost of 0.000163ETH, 0.000244ETH and 0.001085ETH per transaction respectively with the Ethereum's Rinkeby Test Network. Gas price and Gas Limit can be increased or decreased based on the user convenience on how fastly the user wants to complete the transaction.

Using a commodity computer with Intel i3 processor having 4GB RAM and hard disk drive over a 100 Mbps internet connection, the system was evaluated to estimate the number of minimum and maximum transaction that can be performed in a day.

But it also highly depends upon the gas limit and gas price provided for each transaction. Using the RinkeBy test network of Ethereum blockchain with gas price of 10 GWEI(a unit of currency ether and 1ETH(ether) = 10power(9)GWEI) and gas limit of 21000(the maximum number of units of gas which can be used to perform the considered transaction). These values can be changed while performing the transaction. These values have direct effect on the transaction to be written in blockchain. As transaction cost/mining fees is equals to the multiplication of gas price and gas limit. Also, miners picks the transaction with the higher gas price and gas limit. That is why a transaction with higher gas price and gas limit can be written prior to the transaction with low gas price and gas limit which is also independent of the order of transaction performed.

From the analysis of the application we got the above results given in the performance table for both type of transactions in ethereum. As our application has 4 types of push transactions. As soon as the number of transactions happening are within limits of the results shown, the system will perform in fluent manner. As we are also considering the IoT sensors attached with the application which are supporting machine-to-machine types of transcations. for example, detection of faults during power transmission. For all such type of events we had to consider the constraints associated with the scalability of the blockchain. For these types of situations we can consider either other types of blockchain like private blockchain, MultiChainDB which provides a faster rate than the ethereum transaction rate. Also, Vitalik Buterin, the creator of Ethereum, has explained in a recent OmiseGO AMA session that with second-layer solutions such as Sharding and Plasma, the Ethereum network will eventually be able to process 1 million transactions per second and potentially more than 100 million transactions per second which will help in the increase of performance of the application.[8] Other than the performance, all other features of blockchain like robustness, immutability of ledger provides the highly secured environment to the whole critical vulnerable data involved in the application.

8 CONCLUSION

The proposed Blockchain based platform enables users with Smart Grid services where the machines and users have their own Blockchain accounts and the users are able to provision and transact with the machines directly to avail the services. This platform is built on the Ethereum platform with the smart contracts for various events written in Solidity.

The benefits of using this Blockchain based platform is that, it is suitable for secure transactions on Smart Power Grid and at the same time decentralized. The system if free from the burden of prior trust. The system is also resilient, secure, auditable. Moreover, the system is autonomous nature which also enables a decentralized and trustless peer-to-peer network where the peers do not have to need a trusted intermediary for interacting with each other. Since a this system is not controlled by a central authority and all the transactions are verified and validated by a consensus among the peers, the peers do not need to trust each other.

The system is resilient to failures, as it is a decentralized peer-to-peer network with no single point of failure. As Blockchain itself is an immutable and durable ledger, all the transactions once recorded on the Blockchain using this system after a consensus among the peers cannot be altered or deleted. Furthermore, the transparent nature of the public ledger maintained by a this system makes it secure and auditable as everyone on the network knows about all the transactions and the transactions cannot be disputed.

REFERENCES

[1] Brody P. and Pureswaran, V. (2014) Device Democracy: Saving the Future of the Internet of Things, IBM. https://public.dhe.ibm.com/common/ssi/ecm/gb/en/gbe03620usen/GBE03620USEN.PDF.

[2] Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf.

[3] Cryptographic Hash Chain. https://en.wikipedia.org/wiki/Hash_chain, 2018.

[4] Cryptographic Hash Function. https://simple.wikipedia.org/wiki/Cryptographic_hash_function, 2018.

[5] Cryptographic Signatures. https://en.wikipedia.org/wiki/Digital_signature, 2018.

[6] Elliptic Curve Digital Signature Algorithm. https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_Algorithm, 2018.

[7] Filament (2016) Foundations for the Next Economic Revolution Distributed Exchange and the Internet of Things. https://filament.com/assets/downloads/Filament%20Foundations.pdf, 2018.

[8] Inter Planetary File System. https://en.wikipedia.org/wiki/InterPlanetary_File_System, 2018.

[9] Privacy of Transactions. https://en.wikipedia.org/wiki/Privacy_and_blockchain, 2018.

[10] Slock.it. https://slock.it/, 2018.

[11] Solidity Documentation. https://solidity.readthedocs.io, 2018.

[12] TransactiveGrid. http://transactivegrid.net, 2018.

[13] Bahga, A., and Madisetti, V. *Internet of Things: A hands-on approach*. Vpt, 2014.

[14] Bahga, A., and Madisetti, V. K. Blockchain platform for industrial internet of things. *Journal of Software Engineering and Applications 9*, 10 (2016), 533.

[15] Biswas, K., and Muthukkumarasamy, V. Securing smart cities using blockchain technology. In *2016 IEEE 18th international conference on high performance computing and communications; IEEE 14th international conference on smart city; IEEE 2nd international conference on data science and systems (HPCC/SmartCity/DSS)* (2016), IEEE, pp. 1392–1393.