IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 25, No. 6, December, 2020, pp. 737-745
Journal homepage: http://iieta.org/journals/isi

Blockchains: Improve the Scalability and Efficiency of Conventional Blockchain by Providing a Lightweight Block Mining and Communication Algorithm

Subba Rao Peram*, Premamayudu Bulla

Department of IT, Vignan's Foundation for Science Technology and Research University (VFSTR University), Guntur 522213, India

Corresponding Author Email: drpsr_it@vignan.ac.in

https://doi.org/10.18280/isi.250604

Received: 17 April 2020
Accepted: 16 October 2020

Keywords:
blockchain, proof of work, lightweight scalable blockchain, distributed ledger technology, proof of elapsed time, proof-of-stake, and searchable encryption

ABSTRACT

To provide secure and reliable services using the internet of things (IoT) in the smart cities/villages is a challenging and complex issue. A high throughput and resilient services are required to process vast data generated by the smart city/villages that felicitates to run the applications of smart city. To provide security and privacy a scalable blockchain (BC) mechanism is a necessity to integrate the scalable ledger and transactions limit in the BC. In this paper, we investigated the available solutions to improve its scalability and efficiency. However, most of the algorithms are not providing the better solution to achieve scalability for the smart city data. Here, proposed and implemented a hybrid approach to improve the scalability and rate of transactions on BC using practical Byzantine fault tolerance and decentralized public key algorithms. The proposed Normachain is compares our results with the existing model. The results show that the transaction rate got improved by 6.43% and supervision results got improved by 17.78%.

1. INTRODUCTION

Blockchain was first introduced in 2009, in a technical paper by Satoshi Nakamoto titled Bitcoin: A Peer-to-Peer Electronic Cash System. It is an open, decentralized, distributed and time-stamped database containing the entire logged history of transactions in the system. Each block in blockchain contains a cryptographic hash of the previous block, a timestamp and transaction data. These blocks are added at the end of the current blockchain. The correctness of blocks is verified by other nodes using a consensus algorithm, Proof-of-Work in case of Bitcoins. Blockchains are immutable if any transaction is altered in a node's copy the whole blockchain invalidates.

1.1 Features of blockchain

Wherever a manual trust is required, blockchain technology can play a role. Some of the important features of blockchain are:

• Trustless trade - Two gatherings can make a trade without the oversight or intermediation of an outsider, emphatically decreasing or in any event, taking out counterparty chance.

• Durability, dependability, and life span -Because of the decentralized systems, blockchain does not have a main issue of disappointment and is better ready to withstand vindictive assaults

• Transparency and permanence - Changes to open blockchains are freely distinguishable by all gatherings making transparency, and all exchanges are unchanging, which means they can't be modified or erased.

• Process uprightness - Clients can believe that exchanges will be executed precisely as the convention orders expelling the requirement for a confided in outsider.

1.2 Blockchain structure

A blockchain is a distributed decentralized database. It consists of a chain of blocks, new blocks are added at the end of the chain. The blocks once added cannot be altered i.e. a blockchain is immutable in nature. A block in a blockchain is comprised of transaction data, cryptographic hash value of the previous block and timestamp of the time that block is being created. The transaction data is generally stored in form of Merkle Trees [1].

Due to the immutable nature of blockchains, it is used to record transactions whenever trust is an issue between two parties. It provides verifiability as well as permanency of transactions. The distributed network is comprised of multiple peer-to-peer nodes. These nodes follow a commonly decided protocol to communicate with each other and to validate new blocks.

If a given block already added to the blockchain is altered, all the subsequent blocks in the blockchain get altered. Since almost all the nodes in the network have a copy of blockchain, the invalid copy is rejected by the network. Blockchains remove the need of a trusted third party or a central authority by replacing them with cryptographic designs that are able to achieve trust. The Blockchain system mainly consists of the following:

● Block Structure contains two major parts block header and block transactions.

● In the block header, block number, indicates numbers of blocks are created. once value is a random number indicates the freshness of the block. Merkel root is a cryptography hash value of the block transactions using secure hash algorithm 256. A time stamp value and previous header encrypted hash value are also part of the present header file.

● Transactions are identified and few random transactions are created to maintain the balanced binary tree and connected in the Markel tree and generates the Markel root of the present block transactions.

A block consists of cryptographic hash value of the previous block, this forbids any modification to the blockchain. This hash tightly connects each block with the previous block. The following Figure 1 illustrates the phenomena of Block Structure.

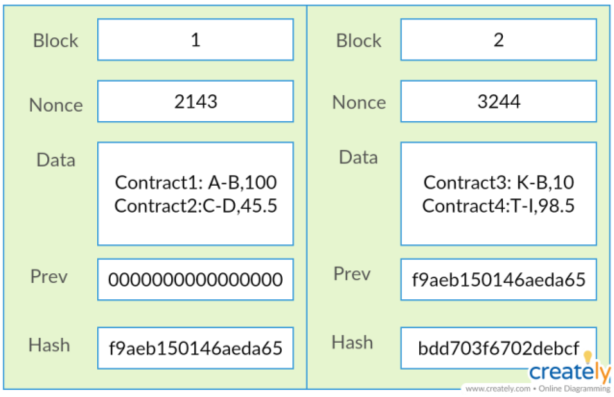


Figure 1. Block structure

1.2.1 Consensus algorithm

Consensus Algorithm refers to the algorithm or protocol that is used to achieve agreement to verify decisions like block creation, new node verification etc. This algorithm establishes the trust in place of the central authority in conventional systems. Some of the consensus algorithms are Proof-of-Work (POW), Proof-of-Stake (POS), Proof- of-Elapsed Time (POET) and Practical Byzantine Fault-Tolerance (PBFT) etc.

1.2.2 Distributed Decentralized Ledger

Distributed Decentralized Ledger which is in the Figure 2 refers to the blockchain itself. The copy of blockchain is stored locally by all the nodes. This ensures that even if some node is compromised, the data remains intact as the data is being held by the complete distributed network and not by individual nodes. The compromised node can be easily detected if it has invalid blockchain [2].

1.3 Blockchain potential applications

Blockchains can be used to automate any application that requires manual trust and thus a third party for verification. some of the potential applications of blockchain technology are:

- Smart contracts
- Achieving security in Internet-of-Vehicles
- Privacy and Decentralization for Peer to Peer Communication
- File storage
- Decentralizing Privacy to Protect Personal Data in IoT Devices
- Prediction markets
- Internet of Things (IoT)
- Transaction management system on IoT E-commerce

1.4 Motivation

Blockchains offer a great potential to automate the conventional centralized or third-party applications. This serves as a future need of almost all the applications. Blockchains have the potential to change the future by revolutionizing any sector it is used in like e-commerce, finance, trade sector etc. But blockchains cannot be applied to these sec- tors due to its limiting scalability and low efficiency. Blockchains for bitcoins ensure trustworthiness by proof-of-work. The block generation process has a huge computational overhead. Moreover, blockchain offers anonymity which ensures privacy of users but at the same time gives a platform for criminal and illegal transactions. There is a need of improvisation in the blockchain system so that it can be used on a large scale IoT networks and restrict illegal transactions in some ways.

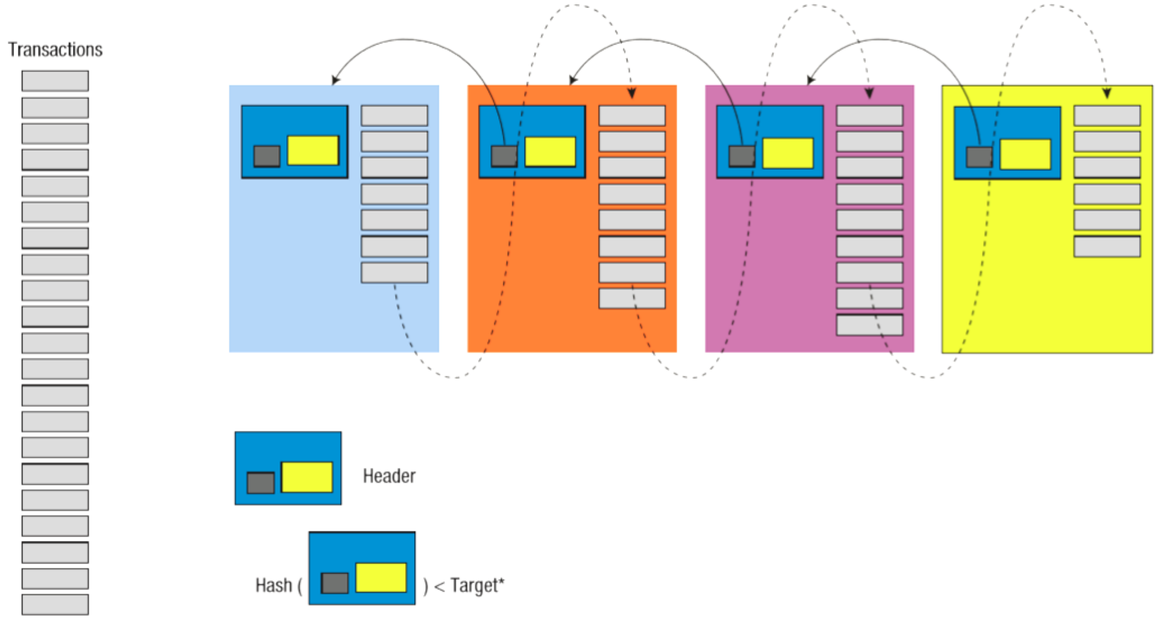


Figure 2. Distributed ledger

1.5 Objectives

The objective of our proposed work are as follows:

- Devising possible solutions to improve the scalability of blockchains so that they can be used for IoT-based applications.
- Finding a possible way of inhibiting illegal transactions without violation of privacy of the users.

In the next session, a literature review of articles providing a solution for the same or related problems is presented.

2. RELATED WORK

2.1 Literature survey

2.1.1 Bitcoin: A peer-to-peer electronic cash system [1]

This is the research paper where blockchains were first introduced. Bitcoins offer an e-cash system that do not require any physical meeting, or personal data sharing between sender and receivers. These transactions need to be recorded in a public distributed ledger called the blockchain shown in the Figure 2. Blockchains record all the bitcoin transactions. Blockchains are comprised of a chain of blocks which are connected using cryptography system. Each block has the hash of the previous block, 0 for the first block. This makes the blockchain immutable to any future changes.

Creating a block requires heavy computation power. The algorithm used for block creation is called Proof-of-work, where each node searches for a nonce by trial and error such that the hash of the block is less than a given target value. This takes around 10 minutes on an average. The heavy computation establishes sense of responsibility and hence ensures correctness of the blockchain. The node that mines the block is rewarded with a certain number of bitcoins, this block reward is halved after certain number of bitcoins have been mined in this way. The total number of bitcoins that can be mined is fixed to 21 Million.

Bitcoins gained great popularity and the value of bitcoins have increased to 6743.65 US Dollars i.e. 4,71,842.27 Indian Rupee.6

2.1.2 Smart contracts: Dumb idea [3]

Any human Interaction requiring trust is recorded in a contract. We have long been using physical contracts and require a trusted third party or central authority for it. There has been a discussion to digitizes this contract system and now blockchains provide a perfect platform for it. The central authority or third party can be completely flushed out of the system. The trust is even greater in blockchains because it is decentralized, the trust and responsibility are a distributed network and therefore if some nodes become dishonest, the trustworthiness can be ensured.

But as it turns out smart contracts can be dumb as human factor is sometimes important in trust. Smart contracts open the possibility in so many sectors but at the same time increase chances of breach. Smart contracts need to be strategically used for the benefit of the humans and business [4].

2.1.3 Decentralizing privacy - using blockchain to protect personal data

Privacy has become a major concern in financial space because of increased reported incidents of financial fraud and misuse of sensitive data. Personal data that is rested in the hands of third parties is not safe because of lack of our control on it. There is a need of decentralized system where owner can decide the authority of control over the data. This decentralized system consists of a secure public ledger and a trusted audit-able computing system using a decentralized network of peers.

Unlike third party, this system acts as automatic access control manager with the combination of blockchain. Complete transparency is ensured. Each user can view, control, access the data which is being collected and can decide up-to what extent it must be shared [5]. This system comprises of three main entities. They area:

- Users:

  Users interact with the system either by downloading or using application.
- Services:

  Application providers, who require personal data processing for targeted ads, personalized services etc. for business processes.
- Nodes:

  Maintaining blockchain and storing distributed private key value data in return for incentives.

This system carries instructions related to sharing data, storing, and querying. All the transactions are not strictly financial but some includes above mentioned processes.

Limitations:

1. Blockchain transactions are highly anonymous which makes using this system for financial purposes challenging.
2. Financial frauds and crimes cannot be tracked and identified.
3. Computational overhead involved solving Proof-Of-Work.

2.1.4 Blockchain: A distributed solution to automotive security and privacy [6]

Blockchains offer to revolutionize many areas. One of those areas is the traffic or vehicular management. Smart vehicles that can auto drive, have GPS, can store locations and much more are already here. Now blockchains can be used to provide security, privacy, and supervision (by whom the owner wants) to the vehicles. This is called internet of vehicles and blockchains can play big role in achieving this.

For security and privacy of automobiles, the automobiles act as the nodes in a distributed network that also contains traffic management centers and some emergency services authorities that are related to crimes, hospitals, or natural disasters. These nodes together with the closest centers form an overlay network that is distributed in nature. One of the powerful nodes in the network is selected as the block manager called the OBM ("Overlay Block Manager") [7].

The nodes in the network, that is the vehicles select their block manager or OBM ac- cording to their location, the one that is closest in distance. The overlay network that these nodes belong to is variable and changes based on their location. These nodes maintain a copy of the data locally in the automobile itself. One back-up of this data is also maintained at the owner's home or any other place of his choice for any case of emergency when the local data gets destroyed. There is automatic alarm system which sends the message to family, friends and to the nearest hospital in case of accident.

This system can be used to collect tolls automatically by connecting a bank account. In case of any illegal activities observed, the location history of the automobile can be demanded from the owner [8].

This system uses public-key encryption system where the user can change the public key or can have multiple public keys for different users that it wants to interact to based on the amount of data they want to share with them. These nodes encrypt the data and sign it with the public key and send it to the OBM. The OBM checks if any node in its network receives data from that key, this data is mentioned in a predefined list that is sent to the OBM. Nodes can periodically update this list or as and when they want to add new public keys to the list.

Once proper public key management, caching of data in the vehicles for emergency and mobility management is achieved, this model can be practically applied to revolutionize the vehicle management system to provide security and privacy to automobile devices and owners.

2.1.5 Peer to peer for privacy and decentralization in the internet of things [9]

- The proposed solution works on the basic idea that the data produced by personal IoT devices are safely stored in a distributed system whose design guarantees privacy, leaving to the people -the real data owners- the decision of which of them to share and with whom.
- This system leverages the use of Peer-to-Peer storage networks in combination with the blockchain.

Personal data that is rested in the hands of third parties poses a greatest risk as they might compromise our data for their self-interests. Goal is to develop a system where data produced by IoT devices is stored in a decentralized distributed system and privacy must be guaranteed [10]. This system leaves the decision of viewing, sharing, controlling, accessing the data to the people who are real data owners. Combining peer to peer networks with blockchain is basic idea behind this system. Three important features of this system are:

1. P2P Network for Data Storage
2. Blockchain
3. Access Policies

P2P Network for Data Storage. Data is broken into several pieces and each piece is stored in different peer. Relying on single peer for data access is useless as it is split and stored in various peers. By setting permissions so that only owner can recompose data present in different peers, privacy is guaranteed by design. Highly robust because of redundancy. Though a peer crashes we can still recover original system [11].

Blockchain. Blockchain is used for two functions.

- certification of data.
- Incentivization of peers.

Access Policies. For the owner to decide different levels of sharing, access policies are combined with P2P network storage. Some of the possible policies:

- Aggregate Data Sharing.
- Obfuscated Data Sharing.
- Raw Data Sharing.

This could be done by employing public key cryptography at application layer. Limitations [12]:

1. Blockchains are not highly scalable. The main barrier to enable a decentralized private-by-design IoT supported by the blockchain is scarce scalability of the present blockchain system.
2. Bitcoin blockchain cannot support high transaction throughput. Transaction speed is not as great as traditional banking systems.
3. Lack of Supervision System.

2.1.6 Normachain

A Blockchain-based Normalized Autonomous Transaction Settlement System for IoT-based E-commerce [13] which refers as Normachains take the case of E-commerce, trade between a buyer and seller where bank acts as an approver. It divides the system into three layers:

1. Transaction Layer
2. Approval Layer
3. Supervision Layer

The workload is divided among these three layers making it more efficient.

Transaction Layer.

- The transaction layer is the layer where actual trade takes place.
- It is comprised of the users that is the buyers and the sellers. Whenever a buyer wants to buy something, they ping the seller node that they want to buy from.
- Both the seller and the buyer node than create the identical copy of a contract and send it to the buyers bank which acts as an approver. The transaction layer does not maintain any blockchain.
- They are free from the overheads of transaction data storage management.

Approval Layer

- The Approval Layer consists of the network of banks.
- Whenever a bank gets a contract from transaction layer, it first requests for authentication from the approval layer that is all the banks as a whole.
- The approval layer uses PBFT algorithm to authenticate the bank.
- Once the bank is authenticated, it verifies the transaction a pushes it to the transaction chain.

Supervision Layer. Normachains solve the problem of illegal transactions by providing a supervision system in place. The top layer of the system is the supervision layer which consists of legally authorized third parties like NGOs, CBI, Supreme Court etc. Anyone cannot join the supervision layer network unlike the transaction layer.

- The nodes in this layer can request for a scan of the transaction for illegal transaction detection.
- These nodes provide a set of keywords to the approval layer which then performs scan on the transaction chain and send the information about the transactions which have those illegal keywords.

This layer structure improves the efficiency. To further reduce the overhead, normachains replace Proof-of-work with PBFT ("Practical Byzantine Fault Tolerance"). Proof-of-work relies on the high computational overhead to ensure authenticity and trust, hence the nodes need to have powerful resources. It cannot be used for IoT based applications. PBFT on the other hand removes the computational overhead. It ensures trust by majority rule. The approver nodes authenticated by all the other approvers or the approver layer before it can verify any transaction and modify the transaction chain [14].

Normachains use searchable encryption that allows search on a cyphertext without revealing the plaintext. It uses public key encryption in a decentralized fashion to ensure privacy while providing a mechanism to restrict illegal transactions.

Normachains propose a solution for IoT based application of blockchains but they have a scope of improvement as follows:

- Normachain stores only one transaction per block. This would result in very large number of blocks as the transaction frequency in e-commerce is very high.
- Public Key Encryption used for encrypting transactions in normachains increases data size, for large number of transactions it would make considerable difference to the block size.

These limitations are addressed in the next chapter to provide an improved version of normachains [15].

3. SCALABLE NORMACHAIN2.0

Based on the existing models for blockchain, we present our new model that utilizes a hybrid approach to provide improved results in terms of transaction speed and scalability [16]. The proposed model of blockchain system for IoT-based E-commerce has following features:

3.1 The three-layer structure of network

The proposed model has three layers which shows in the Figure 3 in the blockchain network viz transaction layer, approval layer and supervision layer as in Normachains [17]. The transaction data is stored at the approval layer in transaction chain. The transaction chain is maintained by the approver nodes [18].

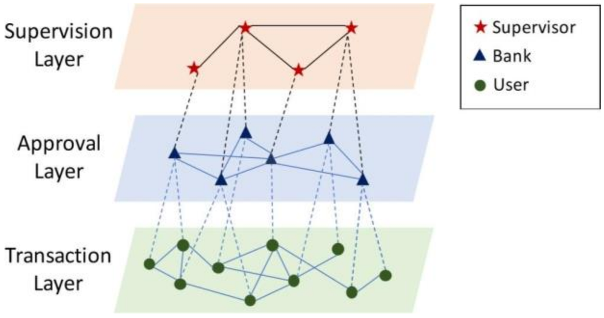


Figure 3. Three-layer network structure

The bottom layer (i.e. transaction layer) is a public exchange layer comprising of clients that are online purchasers as well as Internet business shippers. All client hubs, indicated by Ui, can unreservedly interface and disengage to the organization, as they are not needed to complete any mining or check obligations. At whatever point any client Ui starts an exchange TXi, two indistinguishable agreements between the purchaser and merchant are consequently created and shipped off the center layer's associated banks for endorsement. In this way, the exchange layer does not have any mining liabilities, nor does it need to store the full exchange chain as those obligations are moved to the bottom layer. Clients are just answerable for starting exchanges and sending it to the purchaser's bank for additional preparing [19].

The center layer is a consortium endorsement layer. Nodes in this layer are monetary foundations, signified as node x for banks, which can just check budgetary exchanges for their clients' exchanges. These interconnected devices/nodes validate node x's character and approve him to confirm the produced exchange contract T_x. When an agreement is reached, Bi would thus be able to confirm the exchange contract, encode it with the public key k, and push the ciphertext C_{Tx} onto the exchange chain. Thus, the exchange of every client is just known by his comparing bank and consequently client's protection is not shared to different banks in the endorsement layer. Note that the public key and the private key pair will be produced distributedly. The public key is uncovered to people in general as we need it to encode messages.

The upper layer is a private management layer. All hubs in the management layer are specialists that require a challenge to join. These chiefs, indicated as Si, could be government offices, law implementers, NGOs, and so on Inside a timeframe, the administrators can propose to filter an objective unlawful watchword list w. All the banks will team up to figure the related hidden entrance Tw, if every one of them consider this catchphrase list w is sensible and non-privacy abusing. All endeavors to filter the exchanges are recorded on the management chain to guarantee the responsibility of oversight power. Just if any unlawful data or catchphrases are spotted would it be able to be selected for additional investigation [20].

3.2 Consensus algorithm

The novel practical byzantine fault tolerant (PBFT) algorithm is introduced to achieve consensus for communication among nodes, contract verification and for pushing blocks to the transaction chain. PBFT ensures light weight block mining process as there is no computational overhead as in case of proof-of-work.

PBFT assumes that the maximum number of dishonest nodes does not exceed one-third of the total number of nodes. PBFT is applied to authenticate an approver. If the authentication is granted, the approver is trusted to verify transactions and push new blocks to the transaction chain.

The algorithm used in proposed model is as follows:

```
Algorithm: Identity Verification Algorithm ()
Data: $B_i$: The bank requesting authentication; $B_j$: Other banks in the approval layer.
$|B|$: Total number of banks; $R$: Authentication result
Result: Reurns $R$ i.e the Authentication result to the requesting bank $B_i$.
$B_i$: Broadcasts $REQ_{AUT\ H}$ and $CERT_{Bi}$ ($B_i$'s digital certicate) to all the other banks $B_j$: Verifies if $CERT_{Bi}$ is authentic and broadcasts the decision to banks other than the requesting bank
$B_j$: Receives decisions from $B_k$ where $k\ f= j$
if $R$ is the majority decision i.e $|R| > (|B| - 1)/3 + 1$ then
        return decision $R$ to $B_i$
else
        return $NULL$
end if
```

3.3 Encryption scheme

A decentralized public key encryption with key search (DPEKS) mechanism is used for encryption of transaction data as in Normachains. This ensures the supervision of transaction data for illegal transactions while keeping the security of the users intact. A distributed key generation algorithm is used to generate a public key β and private key infractions α_i. Each approver has the public key and their fraction of the private key α_i.

Therefore, approvers can encrypt the data, but they cannot decrypt any data in the transaction chain individually as they do not have the complete private key. The privacy of users is guaranteed even if at least one approver is honest.

Keyword search is used to search for illegal keywords in the transaction data. Each approver constructs their trapdoor function with their private key α_i and broadcast it. They then form the complete trapdoor by collecting trapdoors received from other approvers.

3.4 Contract verification and block creation process

Whenever a transaction takes place between two users, both the users create identical contracts and send it to buyer's bank. Buyer's Bank authenticates itself, verifies the transaction and stores it in the transaction buffer.

Transaction buffer is maintained at each approver node to store transactions which are not yet pushed to the transaction chain. Buyer's Bank also broadcasts the encrypted transaction to all the other qualified approver nodes, which then update their transaction buffer. Buyer's bank then checks if the transaction buffer is full.

If the transaction buffer is full, buyer's bank creates a block with all the transactions in the transaction buffer and pushes it to the transaction chain. It then flushes it's trans- action buffer. The other approver nodes verify the newly added block.

The algorithm used for block generation is as follows:

Algorithm: Multi-Transaction Block Creation Algorithm

Data: B_i: The bank verifying most recent transaction; B_j: Any other qualied bank; T_x:

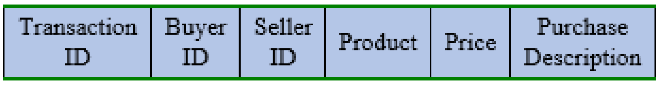
Transaction received by B_i; N: Number of transactions in one block where, $N = 2^k$ preferably, where k is a natural number; $TransactionBuffer_i$: Buffer at B_i of size N where transactions are stored;

Result: Block created if number of transactions in transaction buffer is N.

```
B_i: Receive Contract
if IdentityV ericationAlgorithm() then
        $CT_x = DPEKS(, T_x)$
        Append $CT_x$ to $TransactionBuffer_i$
        Broadcast $CT_x$
        $B_j$: Append $CT_x$ to $TransactionBuffer_j$ $B_i$:
        if Length of $TransactionBuffer_i == N$ then
                Create merkle tree from items in $TransactionBuffer_i$ and store in $TransactionData$
        Create Block from $TransactionData$ Append Block to transaction chain
        end
        $B_j$: Verify Block
end if
```

3.5 Block structure

The contracts in the system consists of buyer ID, seller ID, transaction ID, product, price and description. Users are given a digital ID like in case of bitcoin wallets. Users are represented in the system by this ID only. The personal details of the user relating to a given ID is maintained at that user's local bank only. The bank reveals these details to the supervisors only in the case the user is involved in illegal transactions by maintain the Figure 4 columns.



Transaction ID	Buyer ID	Seller ID	Product	Price	Purchase Description

Figure 4. Contract structure

The contract structure is as in Figure 4. The transaction data is represented in the form of these contracts. Block is formed with N contract units, where N is the block size in terms of number of transactions [11]. N is preferably of the form 2^k, where k is a natural number. These makes sure that the merkle tree does not have unnecessary copies of hash of same data.

Block header will consist of the following data items:

1. Block ID

This is unique for each block and simply is a natural number starting from 1.

2. Approver ID

This is the ID of the approver that mined the given block.

3. Timestamp

This records the time at which the block was created.

4. Previous Hash

Hash of the previous block in the blockchain. This is the link that connects the blockchain. If a given block has wrong transaction data, it will have a different hash than other copies of blockchain in the network. This would lead to all the further blocks in this chain to have different previous hash [21]. Hence the complete chain after the first wrong block becomes invalid.

5. Current Hash

The hash of this block itself. The next block in the blockchain will use this hash as previous hash.

6. Merkle Tree

This is the Merkle tree of the transaction data. The leaves of the tree are the hashes of the transaction data, next level nodes are hash of two hashes combined and so on up-to the root. If any transaction data is changed, the root of the Merkle tree will be changed. Hence invalid blocks can be detected by just matching the Merkle root. Storing transaction data in form of Merkle trees makes block verification efficient and fast.

The block will have the block header and the transaction data. The complete block structure is shown in Figure 5.

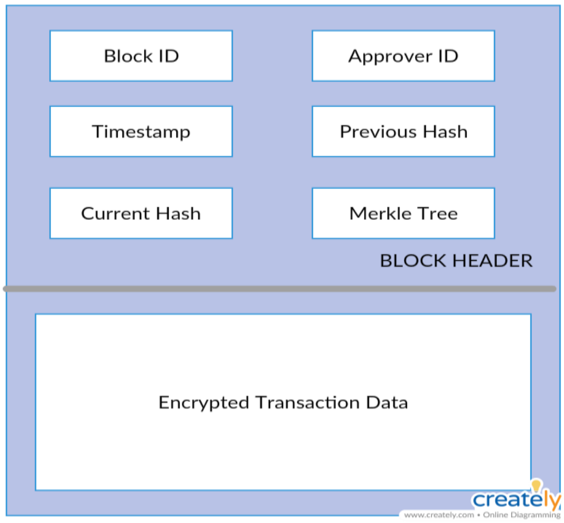


Figure 5. Block structure

4. IMPLEMENTATION AND RESULTS

4.1 Implementation

We implemented Normachains on a laptop with Intel Core i5-4210U CPU @ 1.70GHz, 1701 Mhz, 2 Core(s), 4 Logical Processor(s) and Installed Physical Memory (RAM) of 4.00 GB. The operating system used is Ubuntu 18.04 LTS and programming language C++.

4.1.1 Block structure

The Block Structure we used is as follows:

```
//BlockClass
classBlock{
        private:
        //BlockHeader
        uint64_t block_id;
        uint64_t writer_id;//Approver'sID time_t timestamp;
        charprev_hash[512];
        charcurr_hash[512];
        vector<string> merkle_tree;
        //TransactionData
        vector<string> transactions;
        }
```

Block Header Size:

$$8 + 64 + 64 + 5 + 8 + 448\ \textit{bytes} = 597\ \textit{bytes}\ c\ 0.6\ \textit{Kb}$$

SHA512 hash is used to calculate hashes which have the size of 512 bits or 64 bytes. The number of nodes in the merkle tree are $2 * N - 1$ because this tree is a complete binary tree with *N* leaf nodes where *N* is the total number of transactions in one block.

4.1.2 Transaction data structure

The transaction data has the following structure:

```
//ContractClass
classContract{
        private:

        //ContractDataItems
        uint64_t transaction_id;
        uint64_t buyer_id;
        uint64_t seller_id;
        string product;
        float price;
        string description;
        time_t timestamp;
        }
```

The transaction data is serialized using boost serialization to store it into blocks in form of string. This is done to ensure complex data types like vector and string can be successfully retrieved back after writing blocks to storage files based on items mentioned in the Table 1 [22, 23].

4.1.3 Task division

The task achieved by various nodes is as follows:

Buyer. Buyer requests a purchase from seller and sends a copy of contract to its bank or approver.

Seller. Seller receives purchase request from buyer and returns the price to approve buyer to send contract to the approval layer.

Approver. The Approver nodes receive transactions from buyer nodes. They request authentication from other approver nodes and if successful, add the transaction to transaction buffer. If the transaction buffer is full, they create block and push it to the trans- action chain. They also receive authentication requests or approval requests from other approver nodes, verify the ID of the requesting node and reply with True or False result. They also reply to supervision requests.

Supervisor. They can request the approval layer for scan of a given keyword in the transaction chain and get back the list of transaction ID's in which the given keyword is found.

Table 1. Block header

Data Item	Data Type	Data Size
Block ID	uint64 t	8 bytes
Previous Hash	string	64 bytes
Current Hash	string	64 bytes
Writer ID	string	5 bytes
Time-stamp	time t	8 bytes
Merkle Tree	List of String	64*(2*4-1) bytes

4.1.4 Output

For demo implementation, multiple nodes of type approver, seller, buyer, or supervisor can run on different terminals. These nodes communicate with each other using HTTP protocol which is implemented using socket programming. All the approver nodes must be running at the time any buyer generates a transaction for the transaction to be successfully verified. The same applies for when any supervisor makes a scan request, the approvers must be running to respond to it. When a transaction is generated, the various nodes behaved.

4.2 Results

Execution Parameters are Number of Buyers, Number of Sellers, Number of Approvers, and Number of Supervisors.

4.2.1 Transaction efficiency

For measuring the transaction efficiency, we calculated the time taken to execute one transaction and multiple transactions initiated all together in single buyer thread. The time elapsed was recorded against the number of transactions. Some of recorded the values are presented in Table 2.

Table 2. Transaction results

No. of Transactions	Time: Normachains	Time: Proposed Model
1	0.003383 seconds	0.001736 seconds
10	0.139612 seconds	0.135817 seconds
50	1.80403 seconds	1.61372 seconds
100	5.94251 seconds	5.62581 seconds
200	23.3371 seconds	21.1046 seconds

The plot for total time consumption vs number of transactions is shown in Figure 6 and the plot for averaged time per transaction vs number of transactions is shown in Figure 7.

4.2.2 Supervision efficiency

For evaluating supervision efficiency, we recorded the time taken for completing a keyword search on variable number of contracts. The time elapsed was recorded against the number of transactions. Some of recorded the values are presented in Table 3.

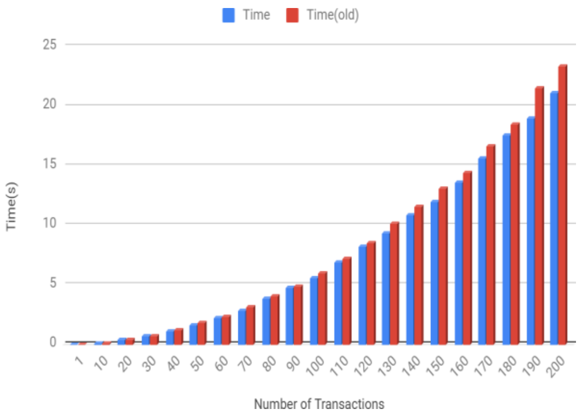


Figure 6. Time Taken vs No. of Transactions

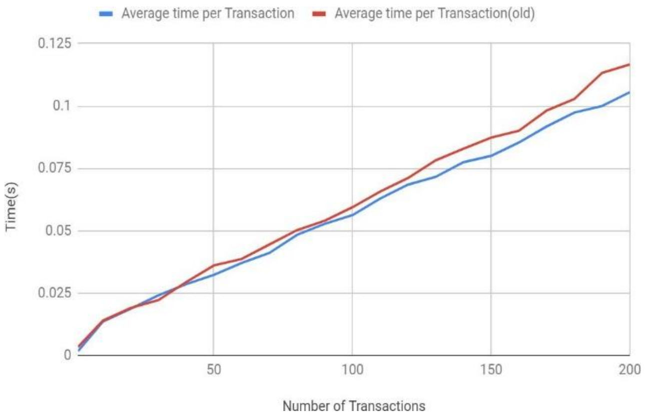


Figure 7. Average Time per Transaction vs No. of Transactions

Table 3. Keyword search on variable number of contacts

Number of Blocks	Time: Normachains	Time: Proposed Model
1	0.230223 seconds	0.191178 seconds
10	2.25801 seconds	1.89706 seconds
25	5.59595 seconds	4.88483 seconds
40	9.02595 seconds	7.20494 seconds
55	12.2747 seconds	9.92576 seconds

There are four transactions per block in this case. The plot for total time consumption vs number of Supervision Results transactions is shown in Figure 8 and the plot for averaged time per transaction vs number of transactions is shown in Figure 9.

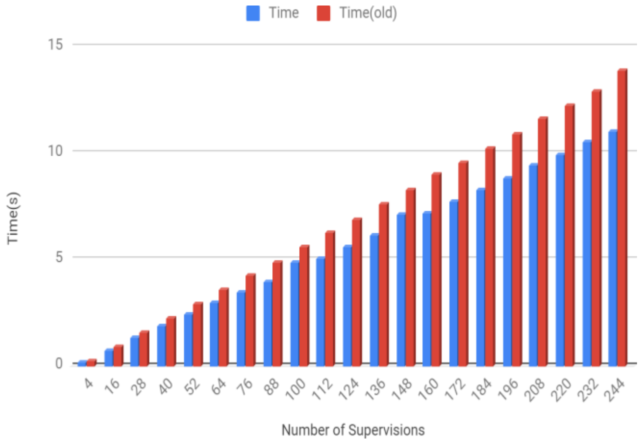


Figure 8. Time taken vs No. of supervisions

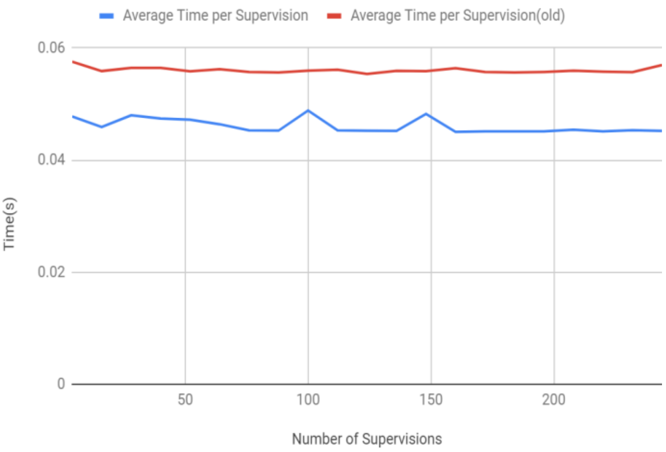


Figure 9. Average time per supervision vs No. of supervisions

In this result, calculated the average of the averaged time per transaction and supervision and calculated the percentage change. The average time per transaction for the available model is 0.06081030203 seconds while the average time per transaction for the proposed model is 0.05690043161 for a block size of four transactions. The percentage improvement is 6.429618486% or 6.43% (rounding off). The percentage improvement for block size of eight transactions is 8.785445299% or 8.78% (rounding off). Similarly, the average time per supervision for the available model is 0.05604767955 seconds while the average time per supervision for the proposed model is 0.04607932468 second for a block size of four transactions. The percentage improvement is 17.78549078% or 17.78% (rounding off) same is shown in Table 4.

Table 4. Percentage improvement

	Time: Normachains (seconds)	Time: Proposed Model (seconds)	Improvement
Transaction	0.060810302	0.056900431	6.43%
Supervision	0.056047679	0.046079324	17.78%

5. CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

In the paper, we have presented a blockchain based model for IoT based applications. The model removes the need of any central authority because of the distributed nature of blockchain. It improves the scalability and efficiency of conventional blockchain by providing a lightweight block mining and communication algorithm and offering division of duties due to its layered structure. It also ensures the privacy of users while putting a check on illegal transactions by offering supervision of transactions using searchable encryption that allows scans for keywords on encrypted data. This model is implemented, and the results show the improved transaction efficiency.

5.2 Future scope

The accuracy of supervision scans is 100% for the proposed model when illegal words are directly used in the contracts. But if they start using argots i.e. different words for words that give away their criminal activities like “cocaine”, “drugs”, “AK 56” etc. then the current system fails because it is directly comparing the keywords provided by the supervisors. This can be achieved by using deep learning along with natural language processing to study the patterns of criminal activities by hit listing some customers previously involved in criminal activities or on current charges against crimes. We plan on achieving this in future.

REFERENCES

[1] Merkle, R.C. (1982). Method of providing digital signatures. Pubication of US4309569A. https://patents.google.com/patent/US4309569A/en

[2] O’hara, K. (2017). Smart contracts-dumb idea. IEEE Internet Computing, 21(2): 97-101. https://doi.org/10.1109/MIC.2017.48

[3] Conoscenti, M., Vetro, A., De Martin, J.C. (2017). Peer to peer for privacy and decentral- ization in the internet of things. In 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), Buenos Aires, Argentina, pp. 288-290. https://doi.org/10.1109/ICSE-C.2017.60

[4] Dorri, A., Steger, M., Kanhere, S.S., Jurdak, R. (2017). Blockchain: A distributed solution to automotive security and privacy. IEEE Communications Magazine, 55(12): 119-125. https://doi.org/10.1109/MCOM.2017.1700879

[5] Eyal, I. (2017). Blockchain technology: Transforming libertarian cryptocurrency dreams to finance and banking realities. Computer, 50(9): 38-49. https://doi.org/10.1109/MC.2017.3571042

[6] Liu, C., Xiao, Y., Javangula, V., Hu, Q., Wang, S., Cheng, X. (2018). Normachain: A blockchain-based normalized autonomous transaction settlement system for iot-based e-commerce. IEEE Internet of Things Journal, 6(3): 4680-4693. https://doi.org/10.1109/JIOT.2018.2877634

[7] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. https://www.researchgate.net/publication/228640975_Bitcoin_A_Peer-to-Peer_Electronic_Cash_System

[8] Zyskind, G., Nathan, O., Pentland, S. (2015). Decentralizing privacy: Using blockchain to protect personal data. In 2015 IEEE Security and Privacy Workshops, San Jose, CA, USA, pp. 180-184. https://doi.org/10.1109/SPW.2015.27

[9] He, Y., Li, H., Cheng, X., Liu, Y., Yang, C., Sun, L. (2018). A blockchain based truthful in- centive mechanism for distributed p2p applications. IEEE Access, 6: 27324-27335.

[10] Taylor, M.B. (2017). The evolution of bitcoin hardware. Computer, 50(9): 58-66. https://doi.org/10.1109/MC.2017.3571056

[11] Herrera-Joancomart´ı, J., P´erez-Sol`a, C. (2016). Privacy in bitcoin transactions: New challenges from blockchain scalability solutions. In International Conference on Modeling Decisions for Artificial Intelligence, Springer, 26-44. https://doi.org/10.1007/978-3-319-45656-0_3

[12] Vukoli´c, M. (2015). The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In International Workshop on Open Problems in Network Security, Springer, 112-125. https://doi.org/10.1007/978-3-319-39028-4_9

[13] Beck, R., Stenum Czepluch, J., Lollike, N., Malone, S. (2016). Blockchain-the gateway to trust-free cryptographic transactions. Research Papers, 153. https://aisel.aisnet.org/ecis2016_rp/153/

[14] Zheng, Z., Xie, S., Dai, H.N., Wang, H. (2016). Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services, 14(4). https://doi.org/10.1504/IJWGS.2018.095647

[15] Dorri, A., Kanhere, S.S., Jurdak, R., Gauravaram, P. (2017). Lsb: A lightweight scalable blockchain for IoT security and privacy. arXiv preprint arXiv:1712.02969.

[16] Dennis, R., Owenson, G., Aziz, B. (2016). A temporal blockchain: A formal analysis. In 2016 International Conference on Collaboration Technologies and Systems (CTS), Orlando, FL, USA, pp. 430-437. https://doi.org/10.1109/CTS.2016.0082

[17] Sharma, P.K., Chen, M.-Y., Park, J.H. (2018). A software defined fog node based distributed blockchain cloud architecture for iot. IEEE Access, 6: 115-124. https://doi.org/10.1109/ACCESS.2017.2757955

[18] Crosby, M., Pattanayak, P., Verma, S., Kalyanaraman, V. (2016). Blockchain technology: Beyond bitcoin. Applied Innovation, 2: 1-35.

[19] Cachin, C. (2016). Architecture of the hyperledger blockchain fabric. In Workshop on Distributed Cryptocurrencies and Consensus Ledgers.

[20] Eyal, I., Gencer, A.E., Sirer, E.G., Van Renesse, R. (2016). Bitcoinng: A scalable blockchain protocol. In 13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16), pp. 45-59.

[21] Swan, M. (2015). Blockchain: Blueprint for a New Economy. O’Reilly Media, Inc.

[22] Bodapati, J.D., Veeranjaneyulu, N. (2019). Facial emotion recognition using deep CNN based features. International Journal of Innovative Technology and Exploring Engineering, 8(7): 1928-1931.

[23] Bodapati, J.D., Veeranjaneyulu, N. (2019). Feature extraction and classification using deep convolutional neural networks. Journal of Cyber Security and Mobility, 8(2): 261-276. https://doi.org/10.13052/jcsm2245-1439.825