**LITERATURE SURVEY**

**TYPES OF SYSTEM PERSENT IN BLOCKCHAIN**

# Decentralized Blockchain

The decentralized nature of blockchain technology means that it doesn’t rely on a central point of control [1].This removes the need for powerful central authorities and instead hands control back to the individual user. A lack of a single authority makes the system fairly more secure. In Decenterlized system relying on a central authority to securely transact with other users, blockchain utilizes innovative consensus protocol across a network of nodes, to validate transactions and record data in a manner that is incorruptible. As the system does not rely on a central authority, the fees that are normally collected by these organisations are no longer a factor. Therefore, transacting on the blockchain can be seen to be cheaper, as the only costs incurred by the parties involved are the nominal fees used to reward the miner or forgers that run a node on the network. The information recorded on the blockchain can be certain to be true as it is near impossible to manipulate due to there being multiple copies that require a complex consensus to be edited. The data is made even more secure by the fact that there is no reliance on a central point of storage, reducing the risk of it being lost or destroyed. Attacking one point of storage would result in no loss of data since all the information is stored on multiple devices around the world.

**Problem with decentralized system**

* **Low throughput**:  decentralized system can only process a limited number of transaction [2].
* **Slow transaction times:** The time required to process a block of transactions is slow. For example, Bitcoin block times are 10 minutes, while Ethereum block times are around 14minuts.
* **Not environmental friendly:** massive amount of miners mining on the network means that difficulty needs to keep increasing thereby leading to mostly useless computations done by miners to out beat each other. It is estimated that every Bitcoin transaction costs about as much electricity that is required to power an average home for eight days. Therefore public ledgers are not very environment-friendly [2].
* **Complex consensus mechanism:** A transaction is introduced to the members, and is only validated, or executed when more than 50% of the network reaches consensus about it. As these transactions have anonymity associated with them. As long as the network agrees that the transaction is valid, the transaction is executed. What if the transaction was from an arms dealer or payment from drug traffickers? There’s very little a government could do to stop the transaction.

# Centralized Blockchain

Centralized platforms require all data (communications/information/etc.) to enter into, and leave through a central hub [3]. That is to say, you physically can’t send or receive any information without it going through that single point, which is often a private server or hub. At the heart of it, blockchain level encryption ensures that your data is safe, in the same way a decentralized blockchain does by ensuring that transactions cannot be altered without invalidating the entire chain. with a centralized blockchain platform, a single point of data collection exists that can identify when unscrupulous activity it taking place; the ability to do this satisfies regulatory and governmental bodies with regard to the problem of consensus. There is one, trackable central point of data dissemination.

**Advantages of Centralized Blockchain:**

* **High throughput:** It can process any number of transaction at any time [3].
* **Data Security:** Blockchain level encryption ensures that your data is safe, in the same way a decentralized blockchain does by ensuring that transactions cannot be altered .
* **More customizable:** Centralized blockchains offer much more customizability and control over the network to the organization deploying it as they can decide who gets to participate in the network.
* **Fast Transaction:** The time required to process transection is too low

**CONCLUSION**

According to our problem statement, we want a system which can process the transaction in a faster manner and it can process any number of the transaction at the given time.it is also easy to update our business logic easily. For our Problem statement, Centralized Blockchain would be a best option.

**REFERENCES**

1.Mirko Kickvic’s Blog:https://medium.com/altcoin-magazine/blockchain-centralization-is-that-the-future-of-decentralized-network-5d95c6f89de6

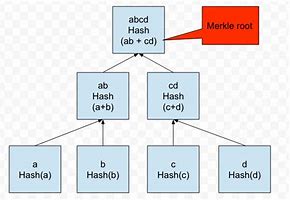
2. BitRewards Blog: <https://medium.com/@bitrewards/>

3. CoinDesk Aricle:https://www.coindesk.com/information/what-is-a-decentralized-application- dapp

**TRANSACTION IN MERKLE TREE**

Merkle Tree: It is a datatype used to store the hash values of multiple transactions in a go. It uses several the transaction’s hash values, preferably in power of two, and create a hash value of it, which is stored in the block [1].

Working: It works in layer-wise format. First layer of consists of hash values of all the transactions. It starts by calculating the hash value of the first two transactions and result will be stored in next upper layer [2]. This continues till the last. If any transaction is left alone, then it is hashed with itself and result is evaluated. Hashing technique is selected before starting of the program.



**Fig. 1:** Construction of Merkle Tree

In fig. 1 transaction of the lowest layer in pair of two is hashed to get the hash value of the next level. This is done in recursive manner until a single block is achieved ie. the merkle root.

The merkle tree reduces the size of the block. It includes a single hash instead of many hashes and occupies low traffic while sending the data.

The merkle tree also involves some disadvantages: Adding new update is very costly. It changes the merkle root value which results in changing the block header value which in turn will break the chain of blocks. We thought of not including the merkle root value while calculating the merkle root value but then we cannot provide security to the stored values.

An Alternative Approach:

SQL: Structured Query Language. It is used to query, monitor and update the database. It is a domain specific language used to manage the data held in the database management system.

Inclusion of an SQL table in each block.

SQL tables are highly manageable. Update is not so costly.

Changes made:

CustomerID will be used to address each block. It will be consisting of the PrevBlock, NextBlock addressing the other adjacent blocks. One block of its own header and SQL table and a block defining the total number of coins each customer has.

Each transaction will be added to the database only after passing the **proof of authority** test. No transactions will be allowed be remove or update once added.

**Comparison:**

* Updates take a lot of computation in Merkle tree whereas it is very easy in case of the database management system.
* Update takes very long time in merkle tree whereas this is not the case in DBMS.
* Since all the data will be verified at start itself using the **PoA** technique, providing security is not required. So, no use of Merkle tree’s security.

**Result:**

Selected DBMS and SQL to store the data in the blockchain.

**REFERENCES:**

1. https://www.codeproject.com/Articles/1176140/Understanding-Merkle-Trees-Why-use-them-who-uses-t
2. https://blockonomi.com/merkle-tree/

**DIGITAL SIGNATURE IN BLOCKCHAIN**

**ECDSA**

This algorithm is used in bitcoin and ethrium [1]. It is an public key encryption algorithm which solves a group of an elliptic curve points. First of all the encryption is done on the basic property of the elliptical curve that it is symmetric along x axis ,the curve used for bitcoin and ethrium is called sect256k1 which in a form of equation is y^2=x^3+7. It has a property that any two point on the curve added together will give third point on the curve and then the symmetric point can also be plotted(symmetric along x-axis). Suppose we have a point p then we can calculate the third point using p+p and so on. The algorithm is used for 256 bit integer so there will be maximum of 510 steps to get the result let x be the integer then total no of combination of the integer will be 2^255 we need to compute xp which will give a result let it be X then we can give the equation as X=xp here we cannot get x using X which satisfies the property of public key encryption and so X can be taken as public key and x be the private key. This also has an advantage that there is no complex computation of division, it can be easily computed just using addition

The disadvantage of this system is the possibility error which makes it possible to select a private key value such that identical signatures for different documents can be obtained. However, enormous computational performance is required for this chance to materialize.

**EGIAMAL ENCRYPTION SYSTEM**

It uses deffie Hellman encryption [2]. It is an asymmetric encryption also ie. different key is used for encryption and decryption suppose there are two user A and B then A forms a mix of its private key and public key (Apub+Apri) and B forms a mix of its private key and public key (Bpri+Bpub)

Then these two exchanges their mixtures of the keys after the exchange of mixture the two add their private key to the exchanged mixture to obtain a secret key. This type of algorithm ensures encryption as well as digital signature.

This solution is employed in public key certificates for the purposes of protecting connections in TLS (SSL, HTTPS, WEB), messages in XML Signature (XML Encryption), and the integrity of IP addresses and domain names (DNSSEC).

The shortcoming of this approach is that the encrypted text length is doubled compared to the initial length this causes longer computation time and tougher requirements for communication channel.

**RABIN CRYPTOSYSTEM**

This algorithm is an extension of a RSA algorithm which can be solved using Chinese Remainder Theorem [2]. In this algorithm first two prime no is then such that both no on division with 4 gives a reminder as 3. Encryption is done as c=m^2modN, therefore there is 4 possible output on decryption which is given as mp=+-c^((p+1)/n)modp and mq=+-c^((q+1)/n)modq which can be solved using chinese remainder theorem and we have to select any one of the 4 outputs.

Though this algorithm has higher operating speed vs RSA it is needed to select necessary message out of the 4 possible outcomes.

**IDENTITY BASED DIGITAL SIGNATURE**

In this algorithm the digital signature is been added based on the unique id of the receiver [3]. Suppose there are different blocks in a block chain each have its unique id, as we are dealing with a private block chain the unique id of every block is known ,if a sender wants to send the data so that only a authorise person is able to access a data , the sender encrypts the data with the unique id of the receiver ,when the receiver decrypts the data it initially compare the id with the unique id of the receiver then only grants the permission to the receiver to view the data ie. if the receiver is not authorise then it will not be able to view the data even if the data is been received. The implementation of this also is very simple and works very well for a private blockchain

**CONCLUSION**

Comparing all the different algorithms such as RSA,ECDSA,EGIAMAL,RABIN CRYPTOSYSTEM,IDENTITY BASED DIGITAL SIGNATURE we came to conclude that identity based digital signature would be the better algorithm to be implemented as it does not need much high computation power and is better suited for the private blockchain authorisation.

**REFERENCES**

1. https://youtu.be/muIv8I6v1aE

2. https://youtu.be/iwCYey4im6Y

3. https://youtu.be/mdxlFwRF4ek

**CONSENSUS ALGORITHMS**

|  |  |  |
| --- | --- | --- |
|  | Permissionless (Public) Blockchain | Permissioned (Private) Blockchain |
| Acess | Open read/write | Permissioned read and/ or write |
| Speed | Slower | Faster |
| Consensus Algorithm | Proof of Work  Proof of Stake | Pre-approved participants |
| Identity | Pseudonymous | Known identities |

**Fig.1.** Classification of Blockchain

Fig.1. enlists the various distinguishing criteria for classifying the types of blockchain [1].The consensus algorithm evaluates the criteria’s and circumstances that are to be reached in order to validate the blocks that are to be included in the blockchain. The consensus algorithm is an outcome of the Byzantine Generals’ Problem which states that any two devices on the decentralised unreliable network cannot completely and indisputably ascertain that they are representing the same data. The Byzantine Fault Tolerance is an attribute which denotes the tough set of defective nodes that are associated with the Byzantine Generals’ Problem. It can tolerate up to 33% defective nodes that is 3f+1 where f is the total number of faulty replicas present in the system [2]. Typically, there are four proofs used to implement the consensus algorithm. The Proof of work (PoW) is a consensus algorithm used in public blockchain like bitcoin and has proven itself to be successful against Sybil attacks [3]. It involves the concept of mining for adding new blocks by solving a hash puzzle. PoW has its own limitations [4]. It makes use of exhaustive computations and consumes lot of electricity for mining. A common security approach is to accept the changes made in the last block only after the transactions are confirmed in the previous six blocks. This takes almost an hour and thus restricts the usability and applicability of the present day PoW Blockchain technologies. In Proof of Stake (PoS) consensus algorithm the block validator constructs a hash from the collection of data indicating the currency owned by the validator and the duration of ownership. The participant needs to put certain crypto coins at stake to verify the transactions. The block validator is then chosen pseudo-randomly based on the participant’s wealth and coins at stake. The hash function is rerun with the latest timestamp as input in case the hash function does not start with a particular number of zeroes [5]. In Delegated Proof of Stake (DPoS) introduced by Daniel Larimer stakeholders elect the members (witnesses) responsible for block creation. The witnesses generate a block only when it is their turn to do so. If the witness fails in producing the block they are removed from future elections. The blocks are produced after every three seconds and the producers are re-arranged after every 21 blocks. Every voter is given an importance according to their crypto coins at stake [6]. Typically the top 20 witnesses are rewarded for verifying the transactions. DPos is significantly faster and scalable than PoS due to lesser number of participants responsible for block creation and validation. Proof-of-Authority (PoA) is a replacement for Proof-of-Work, which can be used for private chain setups, uses a set of “authorities” - nodes that are explicitly allowed to create new blocks and secure the blockchain. The chain has to be signed off by the majority of authorities, in which case it becomes a part of the permanent record. This makes it easier to maintain a private chain and keep the block issuers accountable. PoA is secure, scalable and interoperable.

For e.g. Oracle's Network is open public permission network based on ethereum protocol with proof of authority consensus reached by independent pre-selected validators. Validators are service authorities who secure the network and seal the blocks. They are the known individuals with active notary public license that means that all their information is in the public domain and any third party can check their identity. With no mining or stake required Oracle's network makes smart contract platform cheaper and faster. It makes open networks more affordable for small and medium businesses.

The blockchain node network administrators are responsible for following governance-spawned functions [7]:

* Assigning and retaining roles/permissions that are used to authorize node activity to trusted and capable participants
* Securing public and private keys for authentication and authorization purposes
* Encrypting data via instituted cryptography practices
* Storing rules that represent smart contracts, and for the invoking of these
* Formulating and instituting consensus validation algorithm(s)
* Maintaining a node’s processing history, and its degree of success
* Recording of service level agreements (e.g., performance, uptime), as approved by the node network administrators
* Managing and monitoring of network performance by:
  + Balancing the load among the nodes
  + Detecting rogue threats and malicious activity
  + Monitoring the machine state of the network (e.g., nodes are operational and in-synch)
  + Evaluating a node’s processing performance against any service level agreement measurements

PoA algorithms favour availability over consistency [8]. Proof-of-Authority (PoA) is a new family of BFT (Byzantine fault tolerant) algorithms which has recently drawn attention due to the offered performance and toleration to faults. PoA requires less message exchanges hence provides better performance. Table 1 depicts the differences between the various consensus algorithms discussed so far [9].

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | PoW | PoS | DPoS | PoA |
| Principle | The solution is complex to deduce but convenient to validate. | Higher the stake of a validating node in the network, the more chances and legitimacy it has to validate transactions. | Panel of delegates elected by users of the network monitor the blockchain and propose changes to the protocol which must be approved by the users. | Blocks are validated if signed by a specified quorum of signers. |
| Node Identity Management | Open | Open | Open | Permissioned |
| Energy Saving | No | Partial | Partial | Yes |
| Tolerated Power of Adversary | <25% computing power | <51% stake | <51% validators | <33.33% validators |
| Throughput(TPS) | <100 | <1000 | <1000 | <2000 |
| Scalability | Strong | Strong | Strong | Weak |

**Table 1.** Comparison of various Consensus Algorithms

Two of the main PoA algorithms are AuRa and Clique. Proof of Authority (PoA) is a family of consensus algorithms for permissioned blockchain. PoA algorithms rely on a set of N trusted nodes called the authorities. Each authority is identified by a unique id and a majority of them is assumed honest, namely at least N/2 + 1.

AuRa is the name for Parity's Proof-of-Authority (PoA) consensus engine, the name originally comes from Authority Round (used to be AuRo). It's used in the Kovan network. In AuRa, each block proposal requires two message rounds: in the first round the leader sends the proposed block to all the other authorities, in the second round each authority sends the received block to all the other authorities. A block is committed after a majority of authorities have proposed their blocks, hence the latency in terms of message rounds in AuRa is 2( N/ 2 + 1), where N is the number of authorities. In Clique, a block proposal consists of a single round, where the leader sends the new block to all the other authorities. The block is committed straight away; hence the latency in terms of message rounds in Clique is 1. Such a huge difference between AuRa and Clique is due to their different strategies to cope with malicious authorities aiming at creating forks: AuRa waits that enough other blocks have been proposed before committing; Clique commits immediately and copes with possible forks after they occur. Clique seems to outperform PBFT too, which takes three message rounds to commit a block. So the number of authorities has a linear relationship with the latency (in terms of message rounds) in case of AuRa. If the number of authorities increases the latency (in terms of message rounds) will increase. On the other hand latency (in terms of message rounds) remains unaffected by the number of authorities. The number of validators(authorities) shall be restricted if a blockchain network is using AuRa consensus algorithm.

**AuRa**

AuRa makes use of three parameters:

1. N: the number of nodes
2. f: the number of faulty nodes
3. t: the step duration in seconds

Time is divided into discrete steps of duration t, determined by UNIX time/t. The index s of each step is deterministically computed by each authority as s = t/step duration, where step duration is a constant determining the duration of a step. The leader of a step s is the authority identified by the id l = s mod N. Authorities maintain two queues locally, one for transactions Qtxn and one for pending blocks Qb. Each issued transaction is enqueued by authorities in Qtxn. For each step, the leader l includes the transactions in Qtxn in a block b, and broadcasts it to the other authorities. Then each authority sends the received block to the others (round block acceptance). If it turns out that all the authorities received the same block b, they accept b by enqueuing it in Qb. Any received block sent by an authority not expected to be the current leader is rejected. The leader is always expected to send a block, if no transaction is available then an empty block has to be sent. If authorities do not agree on the proposed block during the block acceptance, a voting is triggered to decide whether the current leader is malicious and then kick it out. An authority can vote the current leader malicious because (i) it has not proposed any block, (ii) it has proposed more blocks than expected, or (iii) it has proposed different blocks to different authorities. The voting mechanism is realised through a smart contract, and a majority of votes is required to actually remove the current leader l from the set of legitimate authorities. When this happens, all the blocks in Qb proposed by l are discarded. Note that leader misbehaviours can be caused by benign faults (e.g., network asynchrony, software crash) or Byzantine faults (e.g., the leader has been subverted and behaves maliciously on purpose). A block b remains in Qb until a majority of authorities propose their blocks, then b is committed to the blockchain. With a majority of honest authorities, this mechanism should prevent any minority of (even consecutive steps) Byzantine leaders to commit a block they have proposed. Indeed any suspicious behaviour (e.g., a leader proposes different blocks to different authorities) triggers a voting where the honest majority can kick the current leader out, and the blocks they have proposed can be discarded before being committed.

**Clique**

Clique [10] is another PoA algorithm employed in Geth [11]. The algorithm executes in time divisions which are recognised by a prefixed order of committed blocks. When a new period starts, a special transition block is broadcasted. It specifies the set of authorities (i.e., their ids) and can be used as snapshot of the current blockchain by new authorities needing to synchronise. While AuRa is based on UNIX time, Clique computes the current step and related leader using a formula that combines the block number and the number of authorities. Most of all, in addition to the current leader, other authorities are allowed to propose blocks in each step. To avoid that a single Byzantine authority could wreak havoc the network by imposing a sheer number of blocks, each authority is only allowed to propose a block every N/2+ 1 blocks. Thus, at any point in time there are at most N − (N/2 + 1) authorities allowed to propose a block. Similarly to before, if authorities act maliciously (e.g., by proposing a block when they are not allowed) they can be voted out. Specifically, a vote against an authority can be casted at each step and if a majority is reached the authority is removed from the list of legitimate authorities. As more authorities can propose a block during each step, forks can occur. However, fork likelihood is limited by the fact that each non-leader authority proposing a block delays its block by a random time; hence the leader block is likely to be the first received by all the authorities. If forks happen, the GHOST protocol [12] is used, which is based on a block scoring approach: leaders’ blocks have higher scores, thus ensuring that forks will be eventually solved.

## Though PoA is best suited for private blockchains it still faces a lot of challenges. First, it strongly lacks decentralization. Second, PoA suffers from the problem of censorship. Third, chances are that signer key/machine may get compromised and lead to the addition of malicious user to list of valid signers.

## Consensus algorithm is the core technology of blockchain, yet the present day research of the consensus mechanism is still in its infant stage. The consensus algorithm specially designed for different scenarios is still very seldom. “How to enhance the performance of the blockchain in a particular scenario?” is still a question requiring further research to be answered.

**References:**

1. Chris Skinner’s Blog: https://thefinanser.com/
2. L.M. Bach, B. Mihaljevic and M. Zagar, “Comparative analysis of Blockchain Consensus Algorithms”, MIPRO 2018, May 21-25, Opatija Croatia, 1545-1551
3. John R. Douceur, “The Sybil Attack”, International Workshop on Peer-to-Peer Systems, IPTPS 2002: Peer-to-Peer Systems pp 251-260
4. Bozhi Wang, Shiping Chen, Lina Yao , Bin Liu, Xiwei Xu and Liming Zhu, “A Simulation Approach for Studying Behavior and Quality of Blockchain Networks”, Springer International Publishing AG, part of Springer Nature 2018 S. Chen et al. (Eds.): ICBC 2018, LNCS 10974, pp. 18–31, 2018. https://doi.org/10.1007/978-3-319-94478-4\_2
5. https://spectrum.ieee.org/computing/networks/how-blockchains-work
6. https://hackernoon.com/the-difference-between-traditional-and-delegated-proof-of-stake-36a3e3f25f7d
7. The Essential Diversity of Blockchain Nodes, Thomas Burke, CACI, International IEEE Blockchain Newsletter, December 2018
8. PBFT vs Proof-of-Authority: Applying the CAP Theorem to Permissioned Blockchain , Authors: Stefano De Angelis , Leonardo Aniello , Roberto Baldoni, Federico Lombardi, Andrea Margheri , and Vladimiro Sassone, Research Center of Cyber Intelligence and Information Security, Sapienza University of Rome
9. Zibin Zheng and Shaoan Xie, Hong-Ning Dai, Xiangping Chen, Huaimin Wang,”Blockchain challenges and opportunities: a survey, Int. J. Web and Grid Services, Vol. 14, No. 4, 2018, 352-376
10. Clique. https://github.com/ethereum/EIPs/issues/225.
11. Go-Ethereum. https://geth.ethereum.org.
12. G. Wood. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 2014