Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

Implementation of Blockchain for Secure Money Transactions

Mrs. J.Verrendeswari
Head of the department, Information Technology,
Rajiv Gandhi College of Engineering and Technology,
Puducherry, Pondicherry, India

Mr. Aameer J, Mr. Wasim Afsal M
UG - Information Technology,
Rajiv Gandhi College of Engineering and Technology,
Puducherry, Pondicherry, India

***Abstract*— Blockchain represents a distributed database or a decentralized ledger predominantly utilized for the secure exchange of digital currency and the execution of transactions. Each participant within the network possesses access to the ledger, which is updated with every new transaction. The Blockchain ledger constitutes a comprehensive collection of all transactions executed historically. This ledger is an ever-expanding, tamper-proof data structure comprising blocks that encapsulate batches of individual transactions. Completed blocks are appended in chronological order. The advent of Blockchain, particularly through Bitcoin, has exerted a profound influence on the global landscape over the past decade, and it is reasonable to assert that this trend will persist, especially as numerous individuals endeavor to surmount the various constraints that hinder blockchains from achieving mainstream adoption. One such constraint is the exorbitant processing and electrical costs associated with the Proof-of-Work consensus protocol. In this paper, we propose an alternative Proof-by-Approval protocol, which represents a more sophisticated iteration of the Proof-of-Reputation protocol, offering enhanced security and a more decentralized framework than its predecessor, albeit at the expense of reduced performance and increased complexity in setup.**

Index Terms—Blockchain, Proof-by-Approval, Secure transaction.

I. INTRODUCTION

With nations moving toward digital transactions and the internet being as centralized as it is right now, many of our net-based activity often requires us to place a good deal of trust in various organizations. While this model of operation has served us well for many years, it does come with some flaws, the most obvious of which is that our data is usually at the mercy of the centralized authority to whom we conform with.

Blockchain solved this issue by providing a decentralized peer-to-peer network that allows its users to carry out transactions and interactions without having to place trust in each other or a centralized authority, the users of the network onlyneed to place trust in the globally accepted mechanism of the system, Blockchain stores the users' data in a distributed immutable ledger. The mechanism of operation of the blockchain ensures that any data that is put in the ledger is permanently embedded into the ledger and cannot be changed. This is accomplished by making all the data in the ledger cryptographically linked to each other and by having the ledger distributed across a large peer-to-peer network where each node both validates and passes the ledger to fellow nodes, and where the network is able to quickly identify any illegitimate ledgers sent by an attacker node and quickly diffuse it, this procedure is done using a protocol called the consensus protocol. There are many different consensus protocols developed for the Blockchain, the one that is most commonly used and accepted is the Proof-of-Work consensus protocol, this is also the protocol that is used by Bitcoin and Ethereum (although Ethereum is moving towards the Proof-of-Stake protocol). The protocol works by requiring anyone who wishes to add data to the ledger to solve a cryptographic puzzle to do so. The cryptographic puzzle, in turn, is designed to be very difficult to solve, meaning that anyone who wants to put a block on the blockchain would have invested in vast amount of resources to do so, thereby repelling any ideas of malicious activity. This process of solving a complex cryptographic puzzle to put data on the blockchain is referred to as mining. Bitcoin was the first to pioneer the PoW protocol and has used it effectively for more than a decade now.

But mining is an incredibly inefficient process which consumes an excessive amount of power. Studies have shown mining can consume as much as 75 TeraWatt Hours of power in a single year. For context, the country of Switzerland consumes about 58 TeraWatt Hours of power in a year. It is for this specific reason that modern Blockchains struggle to see adoption at a large scale. To tackle this disadvantage, many alternate consensus protocols have been proposed including the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

proof-by-stake protocol (which Ethereuem is planning on shifting to).

One other such example is the Proof-of-Reputation protocol. In this protocol, the act of adding a block to the Blockchain can only be done by a verified group of users, these users are usually large companies or the like who have a strong reputation to maintain and would not risk losing this reputation by adding a fraudulent block to the blockchain (which the entire world would then be able to see). An example of a PoR blockchain is the GoChain. The PoR protocol works especially well in private or permissioned networks, for example, a company could deploy a PoR blockchain to maintain a record of public information amongst all of its employees without risk of being compromised while authorizing specific users to verify and add blocks to the Blockchain. Though this approach does start to dwindle a bit towards the centralized system that Blockchain is ultimately trying to work against, as any data has to be validated and thus passed to a centralized group of users who may then again be managed by a central authority (Although this isn't always the case) before it can enter the blockchain, the primary rules of the Blockchain still haven't changed, once data has entered the chain, it cannot be changed by either the authority managing the permissioned users or the user who verified and put that block on the chain.

But centralized control over who has write privileges over the chain while useful in a private organization can be limiting in certain other cases, for example, what about a government office, which takes up the task of approving and managing various extremely important documents that concern a large number of citizens. If a citizen wishes to get a certain document from the office and the office wishes to make record of this, a Proof-of-Reputation based Chain could be used to accomplish this task but it also puts all authority in the hands of the employees of the Government office, which is not ideal as citizens should be allowed to have authority over the documents that pertain directly to them (i.e citizens should be able to validate that their document has been approved from their end as well). To accomplish this, this paper presents a variant of the PoR protocol, called the Proof-by-Approval protocol that is designed to handle this task.

II. LITERATURE REVIEW

A. Proof of Work and Bitcoin

The Proof of Work (PoW) consensus protocol stands as the preeminent consensus mechanism globally, underpinning Bitcoin, which is currently the foremost Blockchain application. PoW was first conceived by Cynthia Dwork and Moni Naor in 1993, yet it did not ascend to prominence until nearly 15 years later when Satoshi Nakamoto elucidated it in their seminal paper on Bitcoin. Remarkably, it was not even designated as "Proof-of-Work" until 1999, when Markus Jakobsson and Ari Juels employed the term in their own publication.

In Satoshi Nakamoto's Bitcoin paper, a comprehensive framework for achieving a functional Blockchain is meticulously delineated. A functional Blockchain, in this context, refers to a production-ready Blockchain that satisfies a specific set of criteria:

An Immutable Ledger: Attaining immutability within a network can be particularly arduous, as there is no definitive method to ascertain whether a subset of nodes has manipulated data within the Blockchain. To circumvent this challenge, the Bitcoin paper advocates for the cryptographic linkage of blocks within the chain, whereby each block encompasses a hash that integrates all the values contained within it along with the hash of the preceding block. This mechanism effectively establishes a hash link that any node within the network can readily validate. Consequently, altering the value of any block in the chain disrupts the continuity at that block, thereby engendering an immutable chain.

A Peer-to-Peer Network: The establishment of a large-scale peer-to-peer network over the internet is imperative. In the nascent stages of Bitcoin, IRC Seeding facilitated the formation of such a network; however, contemporary practices have shifted towards DNS Seeding. DNS Seeding operates by connecting a handful of known clients to which a new client can link. These known clients subsequently introduce the new client to an array of other clients, perpetuating this process until the client is fully integrated into the network.

Network Consensus: Perhaps the most formidable aspect of the protocol resides in ensuring a unified consensus regarding the Blockchain's current state across the network. In Bitcoin, upon the generation of a new block, it is disseminated throughout the network via the Gossip Protocol, with each node accepting only those chains that surpass the length of the one it currently possesses. In instances of conflicting chains, the network awaits the growth of one of the chains (typically, this occurs with the chain that is more widely disseminated throughout the network) and subsequently opts for that chain.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

that chain. This ensures that as long as more than 50 percent of the network remains authentic, the chain cannot be compromised. Another critical element of Bitcoin's consensus mechanism is mining [3].

Mining is a process by which nodes must resolve a complex computational puzzle to generate a block. This puzzle is designed to be challenging to solve yet straightforward to validate. Through mining, Bitcoin achieves three significant objectives [3]:

1. It regulates the rate at which new blocks can be created by adjusting the difficulty of the puzzle. This ensures that by the time a new block is generated, the chain will have synchronized across the network [3].
2. By rendering mining public—allowing any node within the network to mine new blocks—it fosters competition among nodes to mine blocks ahead of others. Since only one block can be appended to the chain at any given time [3],
3. This competitive landscape complicates the efforts of a fraudulent user attempting to inject a malicious block into the network, as they would likely be contending with numerous legitimate nodes striving to mine a block and reap the associated rewards. The resources required for a fraudulent user to successfully add a fraudulent block would be so substantial that it serves as a strong deterrent. (This is in addition to the extensive validations conducted by each node on the chain) [3].

This is also the rationale behind the protocol being termed Proof-of-Work, as each new block effectively serves as evidence that considerable effort was expended in its creation. However, the challenge lies in the fact that this effort is rendered futile if the node does not generate the block first. This phenomenon contributes to the significant wastage of resources inherent in Bitcoin. Nevertheless, Bitcoin's design laid the groundwork for numerous other blockchain applications, such as Ethereum, and some of its foundational principles have even been integrated into the consensus protocol proposed in this paper [3].

Proof-of-Stake

The Proof-of-Stake protocol serves as an alternative consensus mechanism to the Proof-of-Work protocol, devised to address the considerable energy consumption associated with mining. While the Proof-of-Stake protocol is not the sole alternative to Proof-of-Work, it stands out as the most prevalent and arguably the most effective [7].

In Proof-of-Stake, mining is eliminated. In fact, within this protocol, the term used for individuals who create blocks is "minters." This nomenclature arises because, unlike in Proof-of-Work, where every miner engages in a computational contest with every other miner to add blocks to the chain, minters are selected by the network to create blocks. The fundamental principle underpinning this system is that when a new block requires creation, a group of users volunteers to mine the block. Among these volunteers, the network then selects a subset based on specific criteria to generate the block, subsequently rewarding them for their efforts. This means that at any given moment, each block is created by a solitary user, rather than the first among a multitude of competitors, thereby eliminating the wastefulness of computational expenditure [7].

The pivotal aspect of the protocol lies in the criteria employed by the network to select a volunteer or volunteers. In Proof-of-Stake, each volunteer is chosen based on three factors [7]:

1. A predefined degree of randomness [7]
2. A monetary stake contributed by that specific node [7]
3. The duration for which the node has maintained that monetary stake [7]

It is beneficial to conceptualize PoS as a sort of auction, wherein every node can enhance its probability of being selected as a volunteer by staking as many coins of the network as feasible. (However, coin age and randomness also play a role, ensuring that nodes with substantial holdings do not monopolize the network). Similar to the PoW protocol, the PoS protocol can [7]:

1. Regulate the rate at which new blocks are integrated into the network, as it determines who mints the block [7]
2. Foster a competitive environment that renders it exceedingly difficult for a fraudulent user to introduce malicious content. (In this context, the fraudulent user would be vying against individuals with higher stakes and superior coin ages) [7]

All of these factors serve as deterrents to fraudulent users. However, the PoS protocol does face certain challenges; foremost among these is the difficulty of establishing a PoS protocol from inception. Since the coins of the network are generally utilized as stakes, acquiring these coins during the early stages of a blockchain application—when they hold minimal value—is relatively straightforward. Conversely, navigating the landscape of well-established chains presents a far more formidable challenge. Securing a majority of coins in Bitcoin is not merely costly but exceedingly complex, owing in part to the multitude of transactions that would need to be executed [7].

Security and Privacy:

Blockchain wallets use cryptographic techniques to secure transactions and protect user identities. Studies highlight the importance of private keys and multi-signature wallets in enhancing security.

Privacy-focused cryptocurrencies like Monero and Zcash have been discussed for their ability to anonymize transactions.

Efficiency in Transactions:

Blockchain eliminates intermediaries, reducing transaction costs and time. Literature emphasizes the role of smart contracts in automating transactions.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

Challenges and Risks:

Another problem is that unlike PoW, a monetary stake is needed to participate in the protocol. Now keep in mind, that the stake only exists as a measurement criteria and is returned to the user as is, but stakes are made of coins whose value can change. So that means that one would need to buy into the network before being able to participate. Mining on the other hand has no barrier to entry and is so simple to do that one can do it on a phone (not to great effect but it is possible) [7]
Regardless, the PoS protocol does provide a reason- ably effective alternative to the PoW protocol especially if you are a well established chain (which is why Ethereum is shifting towards it) [7]

B. Proof-of-Reputation

The Proof-of-Reputation consensus protocol (PoR) is the protocol on which this paper derives from. The PoR itself however is derived from the Proof-of-Authority protocol (PoA). In PoA, unlike PoW, users have different levels of authorities and not just any user can validate a block. Instead, specific users known as validators are allowed to validate blocks. This makes sense in a private network inside an organization where the organization manages who has access to the network and what kind of authority they have [12].
The PoR protocol builds on the PoA protocol by having the validators be companies instead of individuals. The motive behind this being that companies unlike individuals have less reason to do something malicious since they risk losing the trust and brand value that they have worked for. Thus one could say that a company is putting its reputation at stake every time it validates a block, this is what acts as the security of the network [12].
As such, in order for the protocol to be as secure as possible, the validators must be organizations with a great deal of reputation to put at stake. Large organizations such as Google, Microsoft etc become ideal candidates to be validators in such a protocol. Once a list of validators is established within the protocol, this list is maintained within the Blockchain [12].
We will go into more of the differences between this protocol and the Proof-by-Approval protocol being presented in this paper in the upcoming sections [12].

C. Hyperledger Fabric

Hyperledger Fabric is a special technology that was originally created by IBM, Digital Asset and Blockstream to create and managed private permissioned Blockchains. The technology is a part of a cluster of projects known as Hyperledger which is managed by the Linux Foundation. Today Hyperledger Fabric is an open-source distributed ledger technology which is seeing a great deal of adoption and traction compared to other similar projects. And it has a thriving developer community constantly striving to make the technology better and simpler to use. The current latest version of Hyperledger Fabric at the time of writing of this report is v1.4.2 [2].
Hyperledger Fabric uses an approach similar to something like Ethereum while still possessing a large number of vital differences. Like Ethereum, Fabric uses Smart Contracts (In Fabric, they are referred to as Chaincode) to provide a highly flexible way to add Blocks to the Blockchain. But the similarities end there and moving forward we present a number of key differences Fabric possesses compared to the former which played a vital role in why we chose this framework for the implementation of our protocol and application [2].
Fabric's smart contracts are special compared to other Distributed Ledger Technologies (DLTs) primarily because they can be written using general-purpose programming languages such as JavaScript, Golang, Java or Python (Currently these are the only four languages officially supported but more languages are being added on a regular basis). This was especially important to us since we came from a largely web development background and we had the luxury of being able to use JavaScript (A language which we were already familiar with) to work with Fabric [2].
Fabric is also primarily designed for private Blockchains. This stands in stark contrast to something like Ethereum where the Blockchain is public and anyone anywhere can participate in interacting the chain. In Fabric, the Blockchain can be setup to only accept interactions with authorized personnel and it is even possible to assign roles to individuals such that any interaction with the Blockchain can be controlled and regulated. This means that Fabric can be used by organizations such as banks, schools, colleges, government offices etc [2].
Another extremely important characteristic of Hyperledger Fabric is that it is customizable. Since

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

Fabric was built to accommodate for a myriad of different industry use cases, it is extremely modular. This means Fabric can be used with any consensus protocol whether it be proof-of-work or practical byzantine fault tolerance (both of which will be discussed later on in the literature survey) or in this case, the custom protocol we have made, proof-by-approval. And since Fabric is primarily private and all the users of the system are known and authenticated, there is no need for a cryptocurrency as is the case with Ethereum or Bitcoin [2].

Having a custom protocol means mining becomes optional which in turn allows for Hyperledger Fabric to be highly performant and efficient. The conditional requirement for mining or cryptocurrency also helps reduce security risks and eliminates potential attack points in the system. This also decreases the overall cost required to deploy the system bringing it more in line with normal distributed systems [2].

III. PROPOSED SYSTEM ARCHITECTURE

Let 3.1 Components

- **Blockchain Network**: A public or permissioned blockchain designed for the meticulous recording of transactions.
- **Wallet Interface**: Intuitive digital wallets facilitating the seamless transmission and reception of funds.
- **Smart Contracts**: Mechanisms that automate the validation of transactions and the disbursement of funds.
- **Oracles**: Instruments for the integration of real-world data, such as exchange rates.

3.2 Workflow

1. The user initiates a transaction utilizing a digital wallet.
2. The smart contract verifies the conditions of the transaction.
3. The transaction undergoes validation through a consensus mechanism (e.g., PoS, PBFT).
4. The recipient receives the funds in their digital wallet in real-time or near real-time.

3.3 Smart Contract Implementation

To exemplify the blockchain-based transaction mechanism, a fundamental smart contract was developed in Solidity and deployed on the Ethereum blockchain. This contract enables users to transmit transaction details, which are subsequently stored on-chain and can be queried later. Each transaction triggers an event, thereby enhancing traceability and auditability.

P.S.: Validation and Approval represent distinct processes within the context of this protocol—validation pertains to the assurance that a block possesses all requisite attributes and adheres to the established guidelines necessary for it to be deemed valid, whereas approval involves the verification that the information contained within the block is not fraudulent.

A succinct overview of the operational process can be delineated in three steps (See below for a more elaborate explanation):

A user constructs a block that he or she wishes to append to the blockchain, signs it, and transmits it to an approver via the P2P network (the transmission process may involve randomly selecting an approver or, in this instance, allowing the user to designate a specific approver).

The approver receives the block from the user, validates it by scrutinizing its hash and verifying the signature to ascertain the legitimacy of the user, and subsequently approves the information contained within the block. The approver then signs the block (which serves as an indication of approval) and returns it to the user.

The user receives the block from the approver, verifies that it has not been tampered with, checks the approver's signature to confirm authenticity, and finally submits the block to the blockchain.

In this scenario, the blocks are exchanged between the user and approver through the peer-to-peer network; hashing can be executed using any standardized hashing algorithm (though SHA256 is recommended due to its widespread usage; alternatively, a more robust hashing algorithm such as SHA512 may be employed, albeit with a potential decrease in speed). Signing can be accomplished using methods such as

A more comprehensive elucidation of the process is as follows:

A user creates a block containing the desired data, establishing a 'user' property that encompasses attributes such as 'timestamp'—indicating when the block was initially created by the user—'no,' which designates the nth block generated by this specific user in the blockchain (thus, 1 would signify that this is the inaugural block created by this user intended for inclusion in the blockchain)—and 'name,' which denotes the username of the user and can be utilized for traceability back to the origin.

Vite is a high-performance decentralized application platform that aims to provide ultra-fast and fee-less transactions. It utilizes a Directed Acyclic Graph (DAG) ledger structure to achieve high throughput and low latency, making it suitable for a wide range of decentralized applications. The native currency of the Vite platform is VITE, which is used for paying transaction fees and participating in the network's consensus mechanism. Vite also features a unique snapshot chain technology that allows users to enjoy instant transactions and efficient data management. Overall, Vite offers a scalable and efficient platform for building decentralized applications and conducting fast transactions.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

Metamask is a digital wallet that allows users to securely store and manage their cryptocurrencies. It is a browser extension that integrates with popular browsers like Chrome and Firefox. Metamask also enables users to interact with decentralized applications (dApps) on the Ethereum blockchain. Its interface is user-friendly and provides enhanced security features like password protection and seed phrase backup. Overall, Metamask is a convenient tool for managing cryptocurrency assets and participating in the decentralized finance ecosystem. Polygon is a layer 2 scaling solution for Ethereum that aims to improve scalability and reduce transaction costs. It is a framework for creating interconnected blockchain networks that are compatible with Ethereum. Polygon's native currency is MATIC, which is used for securing the network, paying for transaction fees, and participating in network governance. MATIC can be staked to earn rewards and help secure the network through a process called Proof of Stake. Overall, Polygon and its currency MATIC play a vital role in enhancing the scalability of the Ethereum blockchain.

Metamask wallet

MetaMask is a popular Ethereum wallet that allows users to interact with the Ethereum blockchain and decentralized applications. It is a browser extension that acts as a bridge between the user and the Ethereum network, enabling them to send and receive Ether and ERC-20 tokens, as well as interact with smart contracts.

In terms of security, MetaMask uses a seed phrase (mnemonic) to generate private keys, which are encrypted and stored locally on the user's device. The seed phrase is crucial for accessing the wallet and should be kept secure and offline. MetaMask also offers a password or biometric authentication option for added security. It is important for users to be cautious of phishing scams and only use the official MetaMask website and browser extension to avoid potential hacks or theft of funds.SSL stands for Secure Sockets Layer, which is a standard security technology used for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browser remains private and secure. SSL is commonly used to secure credit card transactions, data transfer, logins, and other sensitive information. It is indicated by the presence of a padlock icon and "https://" in the URL of a website, indicating that the connection is secure. SSL helps protect against eavesdropping, tampering, and data forgery.

Smart contract

```
ragma solidity ^0.8.0;

import "hardhat/console.sol";

contract Transactions {
    uint256 transactionCount;

    event Transfer(address from, address receiver, uint amount,
string message, uint256 timestamp, string keyword);

    struct TransferStruct {
        address sender;
        address receiver;
        uint amount;
        string message;
        uint256 timestamp;
        string keyword;
    }

    TransferStruct[] transactions;

    function addToBlockchain(address payable receiver, uint amount,
string memory message, string memory keyword) public {
        transactionCount += 1;
        transactions.push(TransferStruct(msg.sender, receiver,
amount, message, block.timestamp, keyword));

        emit Transfer(msg.sender, receiver, amount, message,
block.timestamp, keyword);
    }

    function getAllTransactions() public view returns
(TransferStruct[] memory) {
        return transactions;
    }

    function getTransactionCount() public view returns (uint256) {
        return transactionCount;
    }
}
```

In Solidity, smart contracts are used to automate the execution of agreements on the Ethereum blockchain. These contracts are written in a programming language specifically designed for blockchain technology. They can be used to create decentralized applications (dApps) that run exactly as programmed without any possibility of downtime, censorship, fraud, or third-party interference. This makes Solidity a powerful tool for developing secure and transparent systems in various industries.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

possible for transactions to be made via any approver).
The bank would receive details of the transaction from the block. The bank then after validating, approves of the transaction (but doesn't go through with it yet) if it is not fraudulent and sends the block back to the user.
The user would then validate the block and add it to the blockchain where the block's data essentially acts like a receipt on a permanent unhackable ledger compelling the concerned banks to process and complete the transaction.
In a PoR version of the same, the banks would also put the block into the blockchain which for this context would probably work just as fine but our intention was to try and give users a bit more control over their transactions and this would also force banks to be a bit more more careful with the transactions they will be handling.

IV. COMPARISON WITH POR

This protocol is very similar to the PoR protocol. Which might make one wonder how exactly does this compare to the PoR protocol?
The biggest difference that this protocol has to the PoR protocol is that it adds another layer of validation by having users validate approved blocks. In the PoR protocol, the validator (PoR equivalent of approver) can directly add blocks to the blockchain while putting up their reputation as stake. This protocol strives to try and take as much power away from approvers as possible and in the process try and decentralize as much as can be done. By letting the users validate blocks additionally, there is more pressure on approvers to not attempt anything malicious as compared to PoR. But there are situations where the PoR protocol makes more sense, this protocol does not seek to replace or stand as a better alternative to the PoR protocol but simply provides an additional option to anyone who might find it useful.

V. EXPERIMENT RESULTS

A basic implementation of the Proof-by-Approval is written in JavaScript and run on the React vite runtime. Additional libraries that are used in the program are:

- ws: A library that allows for the creation and use of websockets in the program. Websockets will act as the primary way in which peer-to-peer networking is conducted in the program
- crypto-js: A library that provides various crypto-graphic functions. The program makes use of the SHA-256 hashing function provided by this library
- elliptic: A library that provides various encryption based functionality based on elliptic cryptography. The program uses the ECDSA encryption algo- rithm provided by the library. Note. This imple- mentation of the algorithm uses the secp256k1 curve which is relatively safe but it is highly rec- ommended to not use ECDSA at all in real-world applications (even though Bitcoin uses it) and to instead use EdDSA wherever possible. The library does also provide implementation for the same as well.
- express: A library to create http-servers or REST Apis. The program uses this to create a REST Api through which users can interact with the program

The program upon execution sets up 16 REST Apis and 16 websocket servers which all connect to each other and form a Peer-to-Peer network. Of these 16 servers, 13 belong to regular users whose information can be found.

```shell
npx hardhat accounts
npx hardhat compile
npx hardhat clean
npx hardhat test
npx hardhat node
node scripts/sample-script.js
npx hardh
```

- npx hardhat accounts: This command displays the accounts generated by Hardhat in the local Ethereum node.
- npx hardhat compile: This command compiles the contracts in the Hardhat project.
- npx hardhat clean: This command deletes the artifacts and cache directories created by Hardhat.
- Node: Node refers to the local Ethereum node that Hardhat interacts with for development and testing purposes.

Tailwind CSS is a popular utility-first CSS framework that allows developers to rapidly build modern, responsive web interfaces.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

VI. CONCLUSION

Blockchain has proven to be an extremely ground-breaking technology but its implementation main-stream has been hindered due to various limitations. In this paper, we seek to present a protocol that would address one of the major limitations of Blockchain, the high resource utilization, cost and maintenance of the predominant consensus protocol being used. Our protocol looks to allow for Blockchain's continued in- tegration into everyday life with realistic results. In this paper, we show how Blockchain can be used alongside traditional banking systems to enhance their security and improve the transparency of their transactions

REFERENCES

[1] S.Nakomoto, "Bitcoin: A Peer-to-Peer Electronic Cash System". Cryptography Mailing list at https://metzdowd.com. Oct 2008.
[2] Androulaki, Elli, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart et al. "Hyperledger fabric: a distributed operating system for permissioned blockchains." In Proceedings of the thirteenth EuroSys conference, pp. 1-15. 2018.
[3] P. Hooda, "Proof of Work (PoW) Consensus - Geeks-forGeeks", GeeksforGeeks, 2019. [Online]. Available: https:// www.geeksforgeeks.org/proof-of-work-pow-consensus/.
[4] P. Hooda, "practical Byzantine Fault Tolerance,", Geeks-forGeeks, 2019. [Online]. Available: https://www.geeksforgeeks. org/practical-byzantine-fault-tolerancepbft/.
[5] P. Javeri, "Smart Contracts and Blockchains", Medium, 2019. [Online]. Available: https://medium.com/@prashunjaveri/ smart-contracts-and-blockchains-c24538418bf6.
[6] W. Kenton, "Proof of Burn (Cryptocurrency) Definition", Investo- pedia, 2019. [Online]. Available: https://www.investopedia.com/ terms/p/proof-burn-cryptocurrency.asp.
[7] P. Hooda, "Proof of Stake (PoS) in Blockchain,", Geeks- forGeeks, 2019. [Online]. Available: https://www.geeksforgeeks. org/proof-of-stake-pos-in-blockchain/.
[8] T. Marler, "Hyperledger Fabric: Transaction," Medium, 2018. [Online]. Available: Hyperledger Fabric Medium
[9] P. B, "Architecting a Hyperledger Solution - Things to keep in mind," Hackernoon, 2018. [Online]. Available: Hyperledger Fabric Hackernoon
[10] J. H, "Proposed System - Google Docs 1,", Scribd, 2019 [Online].
Available: Proposed System
[11] Moindrot, Olivier, and Charles Bournhonesque. "Proof of Stake Made Simple with Casper." ICME, Stanford University (2017).
[12] Gai, Fangyu Wang, Baosheng Deng, Wenping Peng, Wei. (2018). Proof of Reputation: A Reputation-Based Consensus Protocol for Peer-to-Peer Network. 10.1007/978-3-319-91458- 9_41.
[13] Buterin, Vitalik. "What is Ethereum?." Ethereum Official web- page. Available:
http://www.ethdocs.org/en/latest/introduction/what-is-ethereum.html (2016).