Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

Design and Use Cases of Voting Application using Permissioned Blockchain on Hyperledger Fabric

Sushila Palwe
School of Computer Engineering and Technology
Dr. Vishwanath Karad MIT World Peace University, Pune,India
sushila.palwe@mitwpu.edu.in

Prashant Lahane
School of Computer Engineering and Technology
Dr. Vishwanath Karad MIT World Peace University, Pune,India
prashant.lahane@mitwpu.edu.in

Aditya Verma
School of Computer Engineering and Technology
Dr. Vishwanath Karad MIT World Peace University, Pune,India
avadityaverma15@gmail.com

Pratiksha Thakur
School of Computer Engineering and Technology
Dr. Vishwanath Karad MIT World Peace University, Pune,India
pratikshathakur360@gmail.com

Shantanu Vidwans
School of Computer Engineering and Technology
Dr. Vishwanath Karad MIT World Peace University, Pune,India
shantanuvidwans123@gmail.com

Amogh Deshpande
School of Computer Engineering and Technology
Dr. Vishwanath Karad MIT World Peace University, Pune,India

Article Info
Page Number: 7058 - 7072
Publication Issue:
Vol 71 No. 4 (2022)

Abstract

Elections are an integral part of democracy. Democracy is most important aspect of today's political world, so there is requirement for the unbiased elections to select a authorities or a group of authorities. Elections empower everyone in the country above a certain age group (limit of age varies country-wise) irrespective of their sociodemographic, to participate in election for voting as per their choice. The most common way of election conductions is using ballot paper. With industry 4.0 revolution, the new ways like electronic voting machines (EVM's) are in use for elections. These methods face certain set of challenges such as security threats, efficiency issue, voting by same person many times, cost overhead, chances of system manipulations. In order to address these challenges, we are proposing design and use cases of voting applications using permissioned blockchain technology. Our aim is to prepare system with decentralized voting using Hyperledger Fabric for establishment of network with permissioned blockchain which overcomes the issues of legacy voting mechanisms and support the free, unbiased and supreme election. In this paper we have discussed various use cases demonstrating the various security aspects that could be used to improve security and reliability of the voting systems. Also, To benchmark the proposed system we used Hyperledger Caliper and tested the performance of the system with various chain codes.

Article History
Article Received*: 25 March 2022*
Revised*: 30 April 2022*
Accepted*: 15 June 2022*
Publication*: 19 August 2022*

Keywords— Blockchain voting system, Hyperledger fabric, Permissioned blockchain technology.

I. INTRODUCTION

In India's general elections in 2019, 912 million individuals were eligible to vote, with over 611 million voters casting ballots. It was the largest democratic election ever conducted in the world. This sum roughly corresponds to the present population of the United States. India has a far larger voting population than Europe as a whole. Thus, as voting takes place at a large scale, it is important for the country to be cautious about the voting system it employs as it not only boosts the voting rate, but also ensures fair elections. The elections are meaningful only if the method used for voting is simple, transparent, secure, tamperproof, and trustworthy. For state and national elections, India employs paperless electronic voting machines (EVMs). Contrary to statements made by Indian election officials, these paperless electronic voting devices are vulnerable to hackers and manipulation. As a result, cryptographic methods such as blockchain must be implemented, and a concrete system needs to be added to provide anonymity, security, infallibility, privacy, and reliability. Blockchain technology has shown considerable promise in a variety of fields, including retail, healthcare, and financial applications, due to its capacity to enable record keeping and the irreversibility of data exchange.

Blockchain, as proposed in the paper, is a truly decentralized and peer-to-peer network. A blockchain's information is append-only, which means that updating the blockchain ledger is not possible once a transaction is added to it. In this paper, we look at how a decentralized blockchain system may be used to guide a safe and fraud-proof voting process, as well as provide a massively vast and scalable alternative to India's present voting procedures. Hyper Ledger Fabric is a popular open-source solution for establishing permissioned blockchains. We suggest employing a permissioned private blockchain network with authority that is responsible for restricting network users and monitoring what transactions they may do. Membership service providers, peer nodes, client nodes, an ordering service, and chain code are just a few of the components that make up Fabric. Each component serves a particular purpose and has a varied responsibility. Endorsement, ordering, validation, and committing are the four key steps of the transaction flow.

In this research, we present a Hyperledger fabric network implementation paradigm for a sophisticated permissioned blockchain-based voting system. The article presents a strategy for establishing a powerful voting system in India, from the grassroots to the highest levels. The paper explains how the method works, beginning with a voter presenting his voter card for registration to get access to voting. The CA then verifies the voter and adds him to the network. The voter casts a vote, which is recorded in the blockchain ledger, with the blockchain updating the vote total globally on its network. The voter may verify whether his vote was counted using the QR Code generated by blockchain generated and sent on the SMS or EVM, while retaining the highest level of secrecy and anonymity.

II. LITERATURE SURVEY

In [1], the researchers conducted various tasks to determine performance of blockchain based voting systems using metrics such as population size, the size of blocks, the creation rate of blocks and the speed of transactions. They did these tasks on both permissionless and well as permissioned blockchain

In [2], the researchers discuss how they used Hyperledger Fabric to construct an electronic voting system. The method provided allows for traditional paper voting as well as electronic voting using software. The system uses blind signatures, secret sharing techniques, and an identity mixer in addition to Hyperledger Fabric to guarantee security and anonymity. Furthermore, the researchers provide thorough documentation of the testing that was carried out, as well as performance evaluation.

In [3], the researchers propose a technique in which a Hyper-ledger permissioned blockchain may be used to transfer thousands of transactions per second onto the blockchain, employing every component of the chain code to reduce the burden on the network. The fabric allows client-defined assets to be used with the fabric composer without the need of cryptocurrency such as Bitcoin and Ethereum.

In [4], the researchers claim that their design method allows for integrated voting using both traditional paper voting and e-voting. It explains its implementation on the Hyperledger Fabric platform, describes the solution's architecture, and shows how it works. Because the issue of employing e-voting in both corporate and government voting has yet to be fully resolved, there is still room to improve existing ways and propose new protocols that will make the voting system resistant to various types of attacks. The authors have provided a modified version of a previous voting method and have used a blockchain voting system to raise trust amongst participants. This method enables for both traditional paper voting and electronic voting to be merged.

In [5], according to the researchers, utilizing Use Case Methodology, the system's primary users or actors were isolated and their interactions with the system determined. The voters, miners (universities and public libraries), central authority, candidates, and voters are the system actors. The various steps of an election process, such as pre-election, candidate and voter registration, voting/balloting, tallying, and auditing, were all listed. The actors' interactions with the system were also recorded, evaluated, and used to create the conceptual system.

In [6], the researchers propose a large-scale voting system based on Delegated Proof of Stake, a permission-based protocol (DPOS). The DPOS protocol can be implemented as a small network of nodes with approval from an administration authority. The study used a one-time ring signature scheme to address the issue of voter anonymity and privacy. For recording, managing, calculating, and verifying votes, smart contracts are used.

III. Approach And System Architecture

Citizens in India are entitled to vote if they meet the following criteria, according to the Election Commission of India:

Every citizen who is 18 years old can enroll.

A regular residence is mandatory.

Only one enrollment location is available.

Overseas Indians are presumed to have their primary residence at the address listed on their passport.

The lifecycle of the Hyperledger fabric based permissioned blockchain voting system depends on voter card issued by all the registered voters of the Election Commission of India. The system is complex, interrelated and highly secure. The prerequisites for the system include:

Voter ID and the voting Centre code mentioned on the voter card

An online Registration portal by the Election Commission of India to register the voters.

Blockchain network running into EVMs at every voting center.

EVM must have the public key of the election commission of India, a fingerprint sensor, an image scanner to read the voter ID (checks if the voter ID is valid/Invalid).

Hyper ledger Fabric application API:

Hyper ledger Fabric application API:

This API interacts with the Fabric network using the Gateway peer capability introduced in Fabric v2.4, and it is an outgrowth of the new application programming paradigm released in Fabric v1.4. For Fabric v2.4 and later, the Fabric Gateway client API is the preferred API for application development. The API request is sent to the blockchain server with all the required data.

TLS:

TLS is a cryptographic protocol that secures all the communications over a computer network. It is used for encrypting all the communications across the network. TLS is enabled for authentication and data encryption when communication takes place between the client and peer, peer to peer, peer to orderer,

and orderer to orderer. When a node transmits a message to another node, the recipient should get the exact identical data that the sender transmitted. Over the duration of the conversation, the data should not be altered. This is achieved using the TLS. It encrypts the data sent over the internet using symmetric key cryptography and employs a public key to verify the identities of the communicating parties over the network.

The system architecture of the permissioned blockchain voting system can be demonstrated in the diagram below

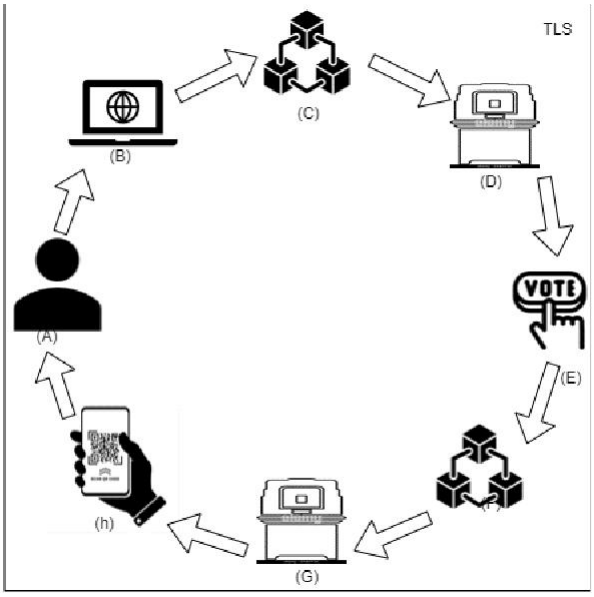

Fig1. Lifecycle of the voting system

Life-cycle of the voting system

A permissioned blockchain network implementation with the help of Hyperledger fabric is mandatory for the administration of the secure voting system. Here we build a blockchain network for every region with the same candidates for election. Each voting center with a center code behaves like an organization in the permissioned blockchain network. The network has fabric certificate authority, Membership Service Provider, endorsing peers, committing peers, chaincode, endorsing policy, ordering service and a ledger that keeps track of all the transactions with respect to the order in which the transactions take place. The diagrammatic flow of the implementation of the whole algorithm represented in Fig (1) goes as follows:

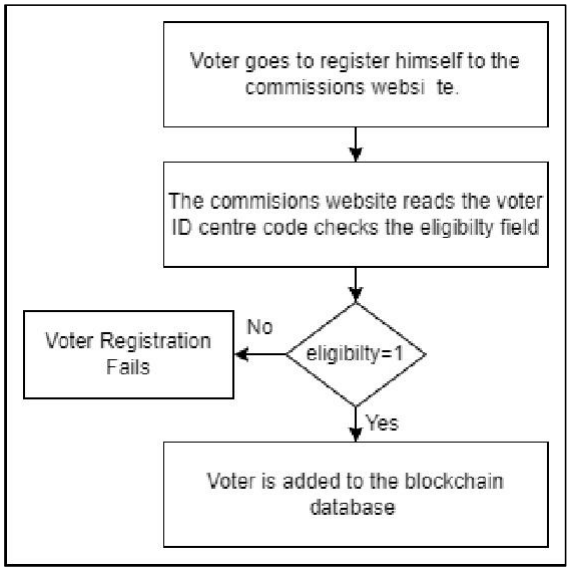

Fig 2. Flowchart of the algorithm for the pre-voting transactions.

The fig (2) shows the algorithmic flow of all the procedure carried out prior to the voting day. The process starts with a voter with valid voting card credentials that include both the voter ID and the center code to register himself on the registration website prior to the election.

a) The registration website has its own database of the voting centers, the voter ID's, and the eligibility status of the voter with binary values of his eligibility to a particular region

b) The eligibility field will be 1 if the voter is eligible to vote, else it will be 0. A query is sent on the website database to check the voter ID and verify the voter's eligibility to vote. Once the voter details are verified, the website sends an API request to the blockchain network. Once the voter registers himself on the website, his registration will be locked on the website for that election.

c) The commission's website sends an API request and adds the voter to the blockchain database i.e., CouchDB (‒voterDB‖). The registered voters are stored in VoterDB channel on the blockchain network. In CouchDB the world state will be updated, and the eligibility of the voter will be marked 1 in the all the organizations of the blockchain network of a particular region. On Election Day, the whole data of the registered voters will be initialized on the ‒VoterDB‖ channel for all the organizations on the blockchain network. The ‒Voter‖ channel will be in an inactive state. Fig (3) depicts the flowchart of the pseudocode and how the algorithm works on the main day.

d) Every EVM must be registered with a voting center. Once inside the voting facility, the voter must scan his voter card with his voter ID visible to the image scanner. After the EVM authenticates the voter ID, the voter is instructed to scan his fingerprint. The voter can now vote the desired candidate.

e) Once the vote is casted, the system employs hybrid encryption to ensure maximum security. The EVM generates a symmetric key that is used to encrypt the voter id. The Election commission's

asymmetric public key is used to encrypt the vote and the fingerprint. The hybrid encryption thus establishes complete voter anonymity as the Election Commission encrypts the vote and does not have access to the symmetric key that encrypts the voter ID.

f) The EVM sends the data to the blockchain chaincode via an API request. The data that is sent to the "vote" chaincode via the API request includes:

Encrypted Voter ID + Encrypted Vote + Unencrypted Voter ID + Fingerprint biometrics + EVM Center Code

Every organization on the network follows the same process. Here we consider the network has two channels viz. "vote" and "voterDB". Every Channel has its own blockchain network and chaincode. The channel Vote has the "vote" chaincode and the channel "voterDB" has "voterDB" chaincode as mentioned in table (1).

Channel	Chaincode
Vote	vote
VoterDB	voterDB

Table (1):Channels with ChainCode

The "voterDB" channel has the CouchDB database "voterDB". The Hyperledger API will request the "vote" chaincode of the "vote" channel. Considering a single organization, first the eligibility is checked of the voter. If the value of the eligibility variable is 0, then the vote cannot be counted, and the transaction will be marked as failed. Then the center code of the "voterDB" of the CouchDB is matched with the center code received from the API request to the "vote" chaincode. If the code matches, the voter is validated. Then the fingerprint is verification takes place. This can be carried out with the help of either EVM or the chaincode. Verification of fingerprints is feasible using EVM, but it has its own security drawbacks like hacking. There are performance constraints if the fingerprints are to be verified using chaincode. However, using an efficient algorithm, the fingerprint verification can be scaled and implemented. Once all the three fields that are voter ID, voting center and fingerprint are verified on the "vote" channel, the transaction gets approved and saved on the blockchain.

g) When the EVM receives the confirmation that the transaction is verified, the EVM executes a second transaction to eliminate the chances of double voting. An update request is passed to the "voterDB" channel of the blockchain that update the eligibility value of the voter that has voted from 1 to 0 in the "voterDB" database. Thus, the eligibility value gets updated throughout all the organizations.

As soon as this transaction takes place, a consensus is achieved from all the organizations. It is impossible for a voter to vote again as he would need to update the CouchDB database of all the organizations in the network. Thus, the possibility of double voting in the EVM is nullified.

h) Once the eligibility of the repeat voter is removed, the EVM generates a symmetric key that had previously encrypted the voter ID. The key will be displayed in the form of a QR code on the EVM to the voter. Using the QR code the user can get to know if the symmetric key exists in the blockchain network. The Election Commission of India can either launch a new website or modify an existing registration website to include technology that reads the QR code created after voting to make vote confirmation process more convenient for voters. The voter can check if the symmetric key is on the blockchain by scanning the QR code on the webpage. This would imply that the voter's vote has been counted. No other information is possible to be displayed through the QR code other than confirmation of voting.

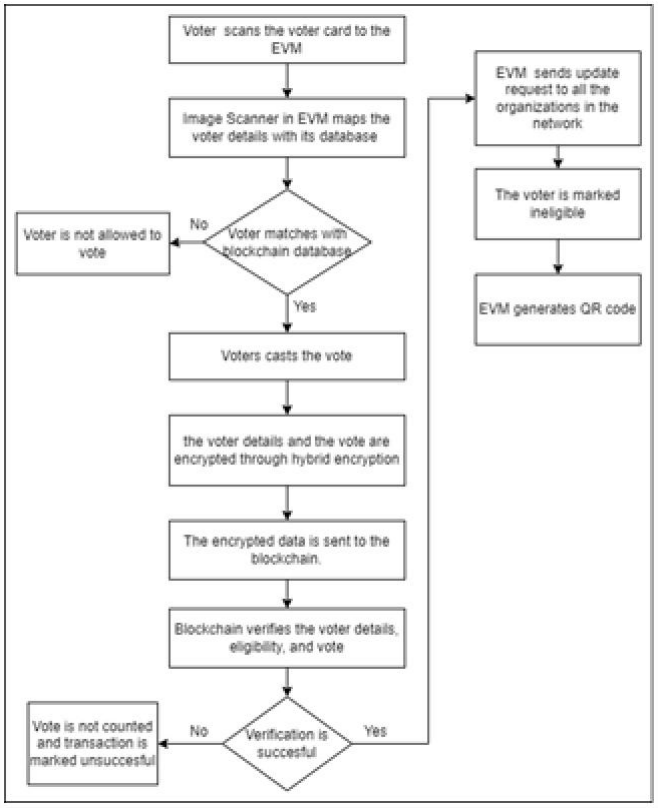

Fig 3. Flowchart of the working model of the post-voting algorithm

IV. Analysis And Results

System's performance was analyzed using a benchmarking tool –caliper‖. Analysis was done for latency, throughput, CPU usage and memory usage, etc. These performance metrics are useful to

understand behavior and response of the system variables to different algorithms used in the chain code and number of transactions carried out.

A. Setup and Specifications

Considered the network with 2 organizations : Org1 and Org2 with a peer to each. As peer0.Org1 and peer0.Org2 and a orderer working as common ordering service.

System Specification:

- System's Platform: WSL backend enabled Docker on Windows 10
- CPU for System : i7 9750H
- RAM for System: 16GB
- Storage Type for System: SSD
- WSL Resource Allocation: 2 cores (4 threads)

Software Specification:

- Hyperledger Fabric with version v2.2.3
- Node Js v16.13.2
- NPM v8.1.2
- Docker v20.10.11
- Docker Compose v1.29.2
- Caliper CLI v0.4.2

Benchmark Specification:

Caliper with Hyperledger Fabric used as Benchmarking tool

- Per second transaction invoked:20,60,100
- Time Periods used (5 different slots) – 10, 30, 60, 120, 240 (in seconds)

Two different benchmark algorithms for the chaincode:

- Basic level: At basic level Transaction are for checking double vote and their prevention without encryption
- Encryption: At encryption level, Transaction are with checking for double vote and preventing that with hybrid encryption of RSA and AES.

The aim of this study is to check the variations in the system's performance when no of transactions increased , algorithms used and transactions received/second.

Following graphs are of System's performance with Hybrid encryption in chaincode and implementation of chaincode without encryption. ‑Base‖ and ‑Encryption‖ are appended with the digit that shows the number of transactions carried per second.

B. Benchmark Result:

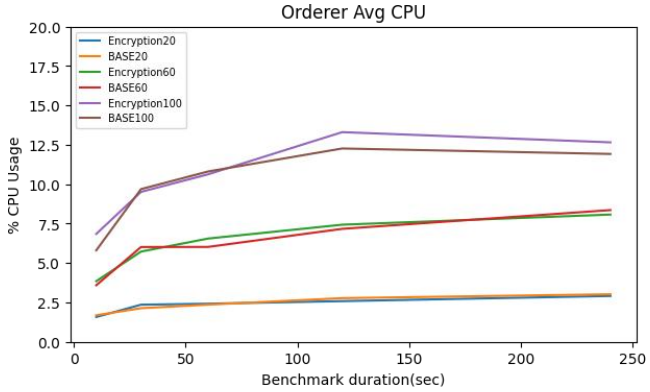

Fig 4Orderer Average CPU Usage

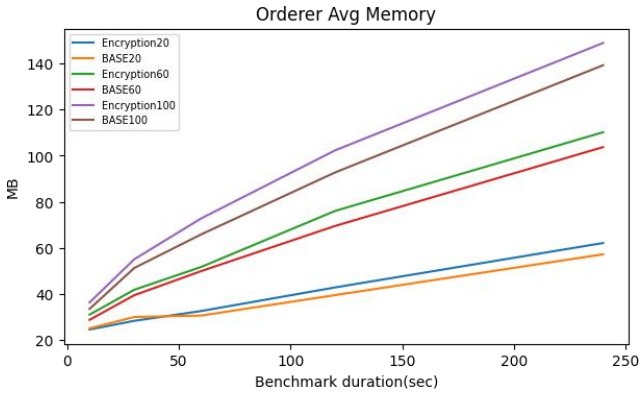

Fig 5 Orderer Avg Memory Usage

In Figure 4. And Figure 5. the aggregate increase in the CPU usage of the orderer is observed as 0.0963 per transactions carried out and the aggregate increase in memory of the ordered observed as 0.5015 per transaction carried out.

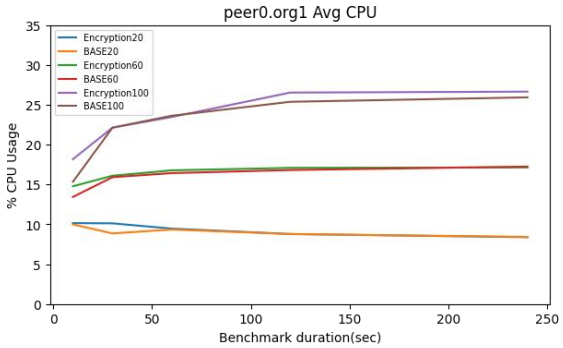

Fig 6 peer0.org1 Average CPU Usage

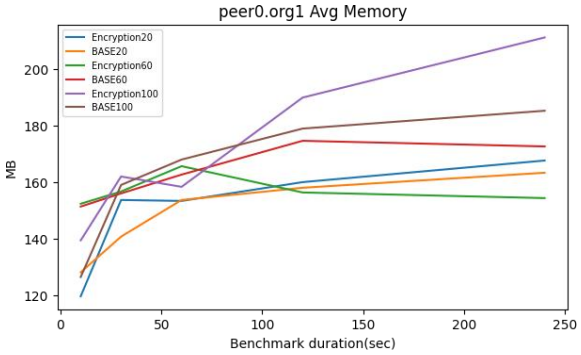

Fig 7 peer0.org1 Average Memory Usage

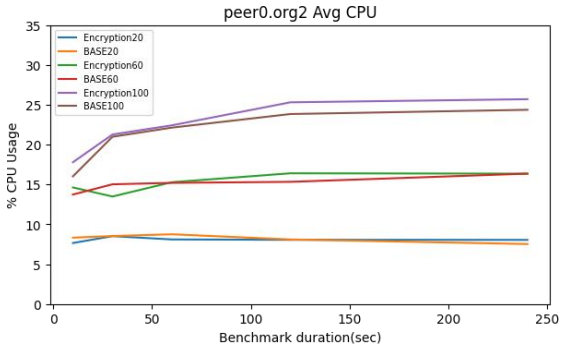

Fig 8 Peer0.org2 Average CPU Usage

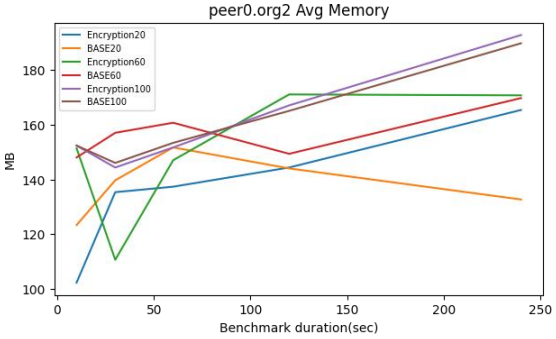

Fig 9 Peer0.org2 Average Memory Usage

With reference to Fig 6 to Fig 9 the consolidated aggregate increase in the CPU usage of the endorsers is 0.166421 per transactions carried out and the consolidated aggregate increase in memory usage was 0.23625 per transaction invoked.

After the analysis of the results obtained from the benchmark following conclusions are drawn

- Number of transactions/sec leads to the increase in CPU usage and memory usage
- Chaincode implementing Hybrid Encryption algorithm observed with increased CPU usage.

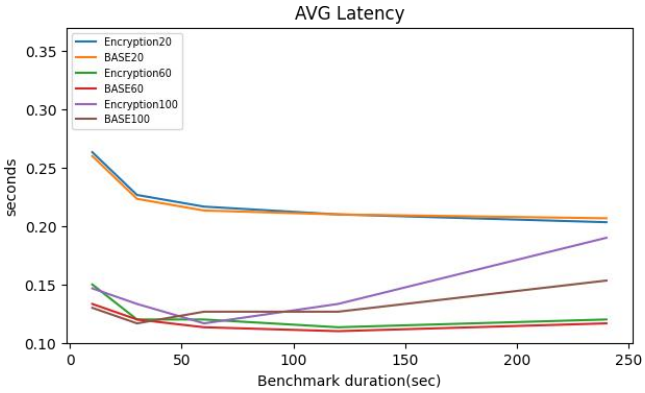

Fig 10 Average latency of each transaction

With Figure 10. It is observed that average latency is reduced with transactions/second carried out increases. With this observation, the conclusion that CPU has to be constant supply of instruction for execution when there is increase in transactions/second. Hence, there is no change in latency/transaction sill the there is reduction in avg latency because of the optimal instruction scheduling with CPU .

The Figure 4. to Figure 10. Are w.r.t the variation in usage of system resources and the average latency of the invoked transactions. The transactions were invoked in Org1 and Org2, the peer0.org1 and peer0.org2 act as endorsers. Fig 4. Shows the memory and CPU usage for peer0.org1 node and Figure 5. Shows the memory and CPU usage for peer0.org2 node Figure 6. shows the average CPU and memory usage as the different ordering service.

V. CONCLUSION

We have added a novel solution to the problems occurring in the current EVM voting system. Properties such as immutability, distributed ledger, and increased security that blockchain provides makes it a perfect solution for creating a voting system. We propose to add additional hardware as well as software components such as a Biometric Scanner and an Image recognition system to the EVM which greatly

increase reliability and security. We have designed the permissioned blockchain voting system in such a way that it eliminates traditional flaws such as double voting, EVM tampering, voter privacy and data integrity. Even when the Election Commission has access to the blockchain data, it never gets to know which voter ID has voted to which eligible candidate due to hybrid encryption.

VI. REFERENCES

[1] Khan, Kashif Mehboob, Junaid Arshad, and Muhammad Mubashir Khan. "Investigating performance constraints for blockchain based secure e-voting system." Future Generation Computer Systems 105 (2020): 13-26.

[2] Kirillov, D., Korkhov, V., Petrunin, V., Makarov, M., Khamitov, I. M., & Dostov, V. (2019, July). Implementation of an e-voting scheme using hyperledger fabric permissioned blockchain. In International Conference on Computational Science and Its Applications (pp. 509-521). Springer, Cham.

[3] Sudershan, K.H., Reddy, P.E., Nadiger, K. and Kumar, S., (2019). Hyperledger based electronic voting system.

[4] Kirillov, D., Korkhov, V., Petrunin, V., Makarov, M., Khamitov, I. M., & Dostov, V. (2019, July). Implementation of an e-voting scheme using hyperledger fabric permissioned blockchain. In International Conference on Computational Science and Its Applications (pp. 509-521). Springer, Cham.

[5] Awalu, I. L., Kook, P. H., & Lim, J. S. (2019, July). Development of a distributed blockchain evoting system. In Proceedings of the 2019 10th International Conference on E-business, Management and Economics (pp. 207-216).

[6] Wang, B., Sun, J., He, Y., Pang, D., & Lu, N. (2018). Large-scale election based on blockchain. Procedia Computer Science, 129, 234-237.

[7] Vidyasree, P., S. Viswanadha Raju, and G. Madhavi. "Desisting the fraud in India's voting process through multi modalbiometrics." 2016 IEEE 6th International Conference on Advanced Computing (IACC). IEEE, 2016.

[8] Garg, K., Saraswat, P., Bisht, S., Aggarwal, S. K., Kothuri, S. K., & Gupta, S. (2019, April). A comparitive analysis on e-voting system using blockchain. In 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) (pp. 1-4). IEEE.

[9] Androulaki, Elli, et al. "Hyperledger fabric: a distributed operating system for permissioned blockchains." Proceedings of the thirteenth EuroSys conference. 2018.

[10] Aghav-Palwe S, Mishra D (2020) Statistical tree-based feature vector for content-based image retrieval. Int J Comput Sci Eng 2020 https://doi.org/10.1504/IJCSE.2020.106868

[11] Aghav-Palwe, S., Mishra, D. (2019). Color Image Retrieval Using Statistically Compacted Features of DFT Transformed Color Images. In: Bhatia, S., Tiwari, S., Mishra, K., Trivedi, M. (eds) Advances in Computer Communication and Computational Sciences. Advances in Intelligent Systems and Computing, vol 760. Springer, Singapore. https://doi.org/10.1007/978-981-13-0344-9_29