Nandagopal et al. 2024. *Int. J. Vehicle Structures & Systems*, 16(4), 522-534
ISSN: 0975-3060 (Print), 0975-3540 (Online)
doi: 10.4273/ijvss.16.4.08

International Journal of Vehicle Structures & Systems
Available online at https://yanthrika.com/eja

Development of Consensus Trust-based Mechanism with Expulsion of Malicious Nodes for Permissioned Private Blockchain Networks

Nandagopal Kaliappan[a], Saumya Chaturvedi[b], R. Parvathi[c], T. Anuradha[d], M. Sathya Priya[e] and Shantanu Datta[f]

[a]*Dept. of Mech. Engg., Dhanalakshmi Srinivasan College of Engg., Coimbatore, Tamil Nadu, India*
Corresponding Author, Email: kaliappan45490@gmail.com
[b]*Dept. of Computer Sci. Engg., School of Engg. & Tech., Sharda University, Greater Noida, Uttar Pradesh, India*
Email: saumyanmishra5@gmail.com
[c]*Dept. of Smart Computing, School of Computer Sci. Engg. and Information Systems, Vellore Institute of Tech., Tamil Nadu, India*
Email: r.parvathi@vit.ac.in
[d]*Dept. of Electrical and Electronics Engg., KCG College of Tech., Chennai, Tamil Nadu, India*
Email: tanura1872@gmail.com
[e]*Dept. of Electronics and Communication Engg., T.J.S. Engg. College, Chennai, Tamil Nadu, India*
Email: sathya77.ashok@gmail.com
[f]*Dept. of Mech. Engg., Asansol Engg. College, Asansol, West Bengal, India*
Email: santanu.datta008@gmail.com

ABSTRACT:

The main objective of this paper is to propose a consensus mechanism, based on trust, for permissioned private blockchain networks. The proposal shows how the use of the cooperativeness of the network nodes is fundamental for the development of a control system, in which the behavior of the nodes is monitored by other nodes. The monitoring of the reputation score is constant in the network and through voting, the node considered malicious to the network is expelled. For the mechanism to work, it is necessary to define rigidity criteria to be applied to actions identified as malicious what are the confidence threshold adopted and what grade receives a node that has not yet been evaluated by the network. Also, determine how many nodes are needed to monitor the behavior of a Malicious Node (MN), in order to identify and expel it more efficiently. The objectives of this work are also to present the performance evaluation of two platforms for the development of permissioned private blockchains in relation to the validation time of a transaction, the response time to chain searches and the transaction mining time and to propose a use case of blockchain technology for application in the storage of electronic medical records in a hybrid approach, which combines the application of public key infrastructure with blockchain technology, to comply with the storage requirements of medical data and offer patient-centric data privacy and access control.

KEYWORDS:

Malicious node; Consensus mechanism; Blockchain; Data privacy; Access control

CITATION:

K. Nandagopal, C. Saumya, R. Parvathi, T. Anuradha, M.S. Priya and D. Shantanu. 2024. Development of Consensus Trust-based Mechanism with Expulsion of Malicious Nodes for Permissioned Private Blockchain Networks, *Int. J. Vehicle Structures & Systems*, 16(4), 522-534. doi:10.4273/ijvss.16.4.08.

1. Introduction

Achieving consensus is a fundamental problem in reliable distributed computing, as it allows participants to coordinate their actions in order to reach common decisions and thus, guarantee the maintenance of the consistency of their states and the progress of the system, despite the existence of failures [1]. In this sense, consensus is fundamental for blockchain technology, as it makes it possible to reach agreement on what information will be added to the chain on the peer-to-peer (p-p) network. There are several Consensus Mechanisms (CMs) that can be used in blockchains to decide on the order of blocks and on which node will be responsible for generating the block. However, the consensus can be seen as a broad mechanism, which allows the network to be self-configuring, according to the cooperation of the participating nodes. It is in the common interest of the participants that the network works safely, since there is no institution administering and controlling the operation of the network. Therefore, the participation of nodes monitoring the behavior of other nodes in the network and the security processes of the blockchain technology allows MNs to be identified and expelled. Thus, from a trust model, the CM can bring robustness to the network and resilience to attacks [2]. In distributed applications, CMs are fundamental for the participating nodes to agree on the operation of the application and the order and veracity of the information that is stored, eliminating the need for a centralizing third party.

In blockchain technology, CMs have the function of determining the order of the blocks and the nodes that were responsible for generation, so that all p-p network

nodes (NNs) agree on the content of the chain and have the replica act - Lizada stored locally. Public blockchain networks usually adopt open CMs, based on mining, where miners compete among themselves for consensus leadership, from a high expenditure of computational power, i.e. Consensus based on Proof of Work (PoW) about cryptocurrency, i.e. consensus based on Proof of Stake (PoS), or other powers of relevance to the miner election that cannot be monopolized, i.e. consensus based on Proof of Authority (PoA). Competition winners receive an incentive, usually in cryptocurrency. The incentive payment is central to the strategy put in place to tolerate Byzantine attacks, such as collusive attacks to subvert the network. Therefore, network governance usually establishes a set of minimum rules for its existence, in addition to criteria for mutual favoring of users [3].

Currently, PoW is one of the few consensus approaches for public networks that are successful and resilient to Sybil attacks [4], where the attacker subverts the reputation of a p-p network by creating a large number of identities. However, in the Bitcoin network, the creation of mining conglomerates has put the network at risk. In addition, the high consumption of energy and processing for the operation of the mechanism generates many criticisms, as it is unsustainable [5]. Private blockchains allow the application of more sustainable CMs, based on voting, thanks to access control mechanisms. However, consensuses based on Byzantine fault tolerance practices face scalability problems, as they generate high message loads for voting [6, 7]. Another option for private networks is the PoA mechanism, in which nodes with permission to generate chain blocks are determined, with the miner function distributed among authority nodes. However, there is no trust mechanism associated with PoA to monitor authority nodes. Thus, the basic concept of technology decentralization is violated, concentrating the power of the network not in one node, but in a group of nodes that can act maliciously.

2. Methodology

In this work, it is proposed to adopt the blockchain network taxonomy in different types of network view: public not permissioned, public permissioned, private not permissioned and private permissioned. Based on this taxonomy classification, the performance of two platforms for the development of permissioned private blockchains, Parity2 and Multichain3, is evaluated. The evaluation consists of comparing platforms, analyzing transaction throughput, block acceptance and chain access latency, with the application of realistic workloads. Workloads are generated by following the probability distribution of transactions arriving on the Bitcoin blockchain. The results show that each platform stands out in specific criteria. Each platform's design decisions result in functionality constraints, which must be addressed by developers to create safer and more efficient chains. In addition, the results obtained with the implementation of the platforms are used as metrics for the development of a permissioned private blockchain simulator [8-11]. Despite the fact that the PoA CM used by the tested platforms is efficient in relation to the operation of the technology, the tested platforms do not maintain a trust mechanism between the nodes. In addition, trusting that a group of authority nodes acts correctly without being for their own benefit, contradicts the security precepts of a blockchain, as it attributes to a group of nodes the authority to intermediate the network transactions. Therefore, a trust mechanism associated with PoA is essential to ensure that Mining Nodes (MiNs) are really trustworthy, based on constant monitoring of their actions [12-14].

Finally, a use case of applying permissioned private blockchains for Electronic Medical Records (EMR) is presented. This is highly confidential information shared between the involved peers to keep the patient's history up to date. Providing privacy to this confidential data is a challenge, because normally after publishing the data the patient loses control over it. So, it is proposed to use blockchain technology for the development of secure EMR applications, where access control is patient-centric [15]. In the proposal, electronic medical records are encrypted on a blockchain, with decryption keys shared only by patients with trusted healthcare professionals. The scalability of the method was investigated and it was shown that the network adapts well, as an increase in the number of nodes (NoN) in the network implies a linear increase in the size of the stored strings. The results show that even with an increased NoN in the network, the time required to introduce new EMRs to the blockchain remains short.

3. Literature review

Rekha et al [16] proposed an authentication and monitoring mechanism in ad-hoc networks, called AMORA, which performs access control and monitoring of NNs without the need for a centralizing entity that authorizes and manipulates the network. This feature brings ad-hoc network applications closer to blockchain network applications. AMORA introduces an authorization for new nodes to join the network from already authorized nodes, forming a delegation chain, trusting the social network among network users to allow new members. In addition, the mechanism proposes the monitoring of actions and the detection of malicious or non-cooperative nodes, so that they do not remain in the network. In the monitoring system, whenever a NN observes a Malicious Action (MA), it contacts the node's delegates, sending a complaint. Delegated nodes check that the report is valid and penalize the node's reputation variable. If it reaches the expulsion threshold, the delegated nodes must issue an expulsion certificate, in which at least a minimum NoN must agree to expel the MN, revoking the network permanence certificate. Ferraz [17] provided a trust-based mechanism for deletion and access control for self-organizing networks to monitor node behavior and exclude underperforming nodes. Access control is obtained through interaction between nodes, along with messages exchanged between witness nodes and referee nodes. Using a scalable trust model, witnesses can interact locally to determine the nature of the defendant and his neighbors from a single hop.

Velloso et al [18] proposed to build a trust model inspired by the trust between humans among nodes in a customized network, called HIT (Human-Inspired Trust Model). Trust levels are based on previous experience and recommendations from other NNs. Past experience allows a node to judge the behavior of other nodes, which can lead to three kinds of judgments, whether the behavior is negative, positive or does not affect other nodes. Virendra et al [19] proposed ad-hoc mobile network security architecture based on trust (Ad-hoc Mobile Networks - MANET). Similar to the work proposed, the goal is to create a measure of trust that allows nodes to make decisions important to the security and proper operation of the network. The architecture defines metrics to evaluate and establish trust between nodes, being a combination of factors. Trust is built by observing the actions taken by the node and recommending it, which is the trust that neighboring nodes have built. In addition to defining metrics, Virendra et al present a trust assessment method that is divided into three phases: initiation and monitoring, search and evaluation, updating and recruitment. These phases address how nodes should act, according to the analysis of trust between nodes for each moment of the architecture.

Zhu et al [20] featured a lightweight authentication protocol for ad hoc networks. It is based on the authenticity of packets transmitted, this characteristic is similar to the validation of a transaction in a chain of blocks, since a node only forwards the transaction to the ahead if it is valid and signed. For ad hoc mobile networks, digital signing using asymmetric keys may require more resources than the devices have. Therefore, the authors offer a signature method that uses a unidirectional keyring for packet authentication and to reduce overhead and establish trust between nodes, this keyring being a list of trusted nodes that can sign the packet using the same key [21]. In blockchain networks, Ongaro et al [22] proposed a CM also based on trust. In Ripple, each NN creates a UNL (Unique Node List). Another CM applied in blockchain is Raft [23], derived from the Paxos protocol [24]. In both mechanisms, consensus takes place in the election of a leader node that is responsible for sharing decisions to be voted. Both mechanisms are vulnerable. In contrast to the proposed mechanism, the proposed trust-based CM is developed for applications on permissioned private blockchains, ensuring distributed consensus. In the proposal, the oldest nodes are willing to become miners according to the maturity criteria. Miners are monitored by a randomly jury that rate the reputation of nodes.

If a miner's reputation falls below a minimum standard, a jury will vote and the miner will be expelled. Additionally, an access control mechanism is proposed that automatically adjusts the network based on the No. of miners required to maintain network scalability. The current contribution is the proposal of a trust-based CM for permissioned private blockchain networks, which offers scalable self-organization of the network, monitoring of MiNs and eviction of MNs. In addition, the performance of two private and permissioned network platforms was analyzed, in which the results are displayed in this paper and were used for the development of a private permissioned blockchain simulator. Finally, a hybrid approach use case of permissioned private blockchain and PKI for electronic medical records is proposed.

4. Trust based CM for blockchain

The collaborative environment is susceptible to attacks and deals with selfish and MNs. Among the most common attacks on public networks is the collusion attack, or "51% attack", in which an individual or institution exceeds 51% of the mining power and can trick the network into accepting illegal transactions or other MAs [25]. In addition, MNs can carry out an eclipse attack in which a node in a strategic position, or a group of nodes in collusion, organize themselves so that part of the network does not have access to the updated chain, filtering transactions and blocks so that the nodes under attack only have access to the attackers' view. If successful, the attacker can mediate most or all of the communication, making it like an eclipse that hides part of the network [26-28]. For example, if Bob makes a purchase through an asset transfer transaction to the seller and after receiving the product or service, the block that stored the transaction is invalidated in a fork, the asset used in the purchase reverts to ownership from Bob. On the other hand, one of the advantages of applications in private networks is to allow the eviction of nodes that have harmful behavior to the network. However, after analyzing the permissioned private platforms parity and multichain, it is observed that the PoA CM is not completely distributed [29]. Mining is under the responsibility of a subgroup of authority nodes, but there is no monitoring of miners' behavior. It is a naive approach to assume that nodes given the miner permit are not subject to failure, hacking or acting on their own behalf. However, in the two evaluated platforms, a mechanism for revoking permission or expelling authority nodes was not observed.

In this work, a trust model is proposed to be associated with CMs in private permissioned networks. The proposal is that, through distributed monitoring, the MiNs are evaluated based on their actions, building a reputation among the NNs. Therefore, for MiN to be maintained, the reputation built must be higher than the trust threshold. If the miner's reputation is below that threshold, a jury made up of NNs votes for expulsion and, if enough votes are accumulated, the MN is expelled from the network. In addition, the model proposes a criterion for the selection of MiNs autonomously and which keeps the network scalable. It is also proposed the creation of a chain of control blocks that follow the same trust criteria associated to the chain of blocks of the application, for the storage of the transactions of CM. In this way, the control data does not mix with the application's blockchain data. In permissioned networks, the distribution of miner permissions can follow different metrics. This proposal follows the concept of network maturity [30]. The maturity criterion can be used as a confidence criterion, as the nodes need to remain actively in the network to be considered candidates for the position of miners. The more time as a network participant, according to its entry

timestamp, the greater its chances of becoming a MiN, thus acquiring the role of block generator for the application and control block chain.

In addition, it is also your responsibility to authorize access and grant miner permission to new nodes, according to the needs of the network. Therefore, the proposed model provides self-organization of the network, so that the growth or expulsion of MiNs does not affect its operation. The trust model follows the scalable monitoring criterion proposed by [17], in which, instead of all nodes evaluating the behavior of a given node, a subset of nodes is pseudo randomly selected to be responsible for monitoring [31]. In this proposal, the subset is called the jury of a MiN and is responsible for evaluating the reputation of the miner according to the observed actions. Each judge assigns an initial score to the MiN and according to the actions performed, this score is updated. If a miner's reputation is below a trust threshold, the jury must vote to evict the untrusted node. Therefore, the model presents a distributed, self-organizing and scalable access control mechanism and a trust analysis to select monitor nodes, monitor reputation and expel MNs. Identifying and classifying the severity of MA is not within the scope of this work.

4.1. Block chain network access control and selection of MiNs

P-p network access control mechanisms may vary according to blockchain application. In this proposal, access control is based on the authorization of at least one MiN, which signs authorizing the entry transaction of the user who wants to participate. The user requests to join a group and expects to be accepted by the group [32, 33]. Fig. 1 shows the proposed access control mechanism. In (i) the node must generate an asymmetric key pair. These keys are used to sign the node's actions on the network and the public key (PK) of the node is used to address it, since there is no certifying authority that maintains the domain of participating nodes [34]. Therefore, the PK of the node must be sent to the network in the form of a transaction, ticket request transaction (Txingress) as shown in Fig. 2(a). In (ii) the transaction needs to be validated, by one of the MiNs that signs the transaction and forwarded to the repository of valid transactions. After mining the transaction in (iii), the transaction is part of a control chain block and all NNs can add it to the list of identification keys, with the timestamp of the ingress transaction. As this is a new node to the network, it must join as a regular node, with access to the complete content of the chain and only permission to generate transactions. Therefore, in this proposal, no confidence ratings are assigned to their actions. To maintain network efficiency and not overload the MiNs, the No. of MiNs M is directly proportional to the NoN in the network N. Therefore, when considering the growth of the network, the No. of MiNs must grow proportionally to the NoN in the network growth and thus avoid scalability problems. According to the maturity criterion [35], the oldest nodes, according to the entry time record, have preference in the selection of MiNs. For this reason, in this proposal, the information on the total NoN participating in network N is stored locally and updated at each entry or expulsion of nodes.

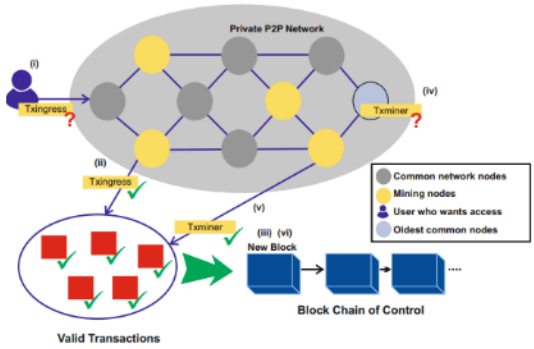


Fig. 1: A mechanism for controlling access to a private P2P network by introducing new MiNs.

The self-selection of miners happens when the set of miners needs more nodes to maintain the network scalability standards, ME M, where ME is the expected No. of miners. In (i), the user who wants access must issue a text message and address the network. In (ii), the MiN allows access by signing to Txingress and sending a valid transaction to the warehouse. Transactions are extracted in (iii) and the entire network is aware of the new nodes. In (iv), the oldest public node is issuing Txminer. In (v), the mining contract is signed and verified by Txminer. In (vi), the entire network is aware of the new MiNs. Calculating the difference between the expected and actual value of MiNs, V = ME- M, we find the No. of vacancies for new miners. Recording the time when the node joined the network makes it possible to sort the entry transactions and allows all nodes to have single views of which are the oldest nodes. Thus, the αV oldest common nodes must volunteer and issue a request transaction to become a MiN (Txminer), thus preventing nodes that are not active or that do not volunteer to be miners from hindering the scalar growth of the network. Process (iv) shows when a node is ready and issues a transaction type Txminer. Transactions must be validated according to the timestamp to ensure that only the αV oldest nodes apply. In the process (v), the transaction is signed by a MiN, which validates the transaction and forwarded to the valid transaction repository. In (vi), the miner that generates the next block has the task of selecting the Txminer transactions that were issued by the oldest nodes to determine which are the new V MiNs in the network.

Fig. 2(a) presents the information that is stored in the Txingress transaction. The transaction is sent by the user who wants to enter and needs to be verified by the MiN. Fig. 2(b) as new nodes enter, other nodes must become miners and maintain network scalability. Introduce the common node problems in Txminer. In the first moment, the user sends his PK, timestamp of the ticket request and the applicant's signature to the network. After one of the network miners grants access to the new node, in a second moment, the miner signs and adds the PK of the incoming node, the timestamp, the granting of access and the signature of all Txingress content, the transaction Txminer. Similar to the join process, the Txminer is issued by the old node, which sends the PK, timestamp of the miner request and the requester's signature. As the need for new MiNs is common knowledge, a miner must validate the Txminer by adding the PK of the miner, timestamp of granting the

miner permission and signature of all content. Once the transaction is part of a mined block of the chain of control, all nodes agree on the new MiN and this will be monitored by judge nodes (JNs) that will assign grades to all actions and behaviors performed in the network process.

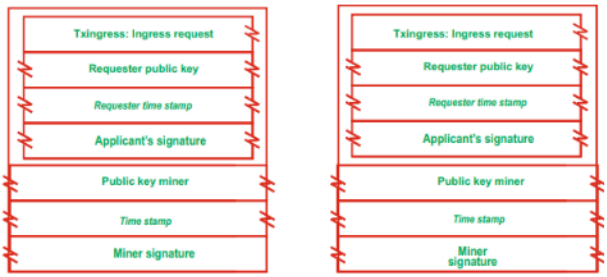


Fig. 2: Transactional access control mechanism, (a) Ticket request transaction and (b) Request transaction to become a miner

4.2. Selection of judges

In the proposed mechanism, all nodes participating in the network are monitored by others, called JNs. Each miner must receive a JE Number of Judges (NoJ), responsible for monitoring the behavior and judging the confidence of the node with each action performed. NoJ needed for a good functioning of the network depends on the expected resilience, considering factors such as the number of MNs, expected time for the exclusion of a MN and the accuracy of the JNs in relation to the trust placed on the defendant node. A node may not be able to identify its judges in advance, so a non-reversible pseudo-random selection algorithm is chosen. Based on the implicit construction of a BF, applied to the PKs of the MiN, the nodes that are determined by the filter assume the position of judges. This filter allows estimating an expected value of judges by changing the filter size and the number of interactions needed to reach an average NoJs. The BF [6] is a data structure used to represent the membership of elements to a set $F = \{f1, f2, \ldots . fn\}$ of n *e*lements. It consists, in general terms, of a vector of m bits and of k independent hash functions (HFs) $h_1, h_2, \ldots ., h_k$ whose outputs vary uniformly in the discrete space $\{0, 1, \ldots, m-1\}$. This vector m is initially composed of all bits equal to zero. k HFs are applied to the elements.

For each element $s_i \in S$, the bits of the vector corresponding to positions $h_1(s_i), h_2(s_i), \ldots ., h_k(s_i)$ are filled with 1, where the same bit position can be filled more than once. Given that perfectly independent and random HFs are used, the probability p that a bit remains zero after insertion of n elements as per,

$$p = (1 - \frac{1}{m})^{kn} \approx e^{\frac{-kn}{m}} \quad (1)$$

where n = the elements contained in the filter, k = the number of interactions hash over the element and m = the length of the filter (bits). In the proposal, the hash of the miner's PK is interpreted as the BF, in which the m bit vector is represented by the selection of the first m bits of the hash. Since the HF is perfectly random, the probability of occurrences of $0's\ and\ 1's$ within the selection of the first m bits is $p = 0,5$.

When considering the creation of the filter implicitly, by the random selection of the PK, a value is observed of expected J_E judges for each miner. In order to commit less processing in k different iterations of hashing, we determine the value of k and the Number of elements $n = J_E$ that belong to the set in Eqn. (3), by applying the probability $p = 0.5$ in Eqn. (1) and the length of the filter is established as,

$$m = -\frac{n*k}{ln(p)} \text{ and } m = -\frac{n*k}{ln(0.5)} \approx \frac{n*k}{0.693} \quad (2)$$

Fig. 3 shows the creation of the filter carried out by all the NNs. From the hash of the miner's PK in step (i), the value of m delimits the length of the filter from the first m bits resulting from the HF in step (ii). Therefore, the membership filter of the JNs must be created implicitly, in which the false positive criterion is not applied. The nodes belonging to this miner's judge set are unknown when the PK of the miner is randomly created. With the formation of the filter, it is the responsibility of all NNs to verify their membership in step (iv), which the node performs k different hash interactions on its own PK to verify its membership to the filter.

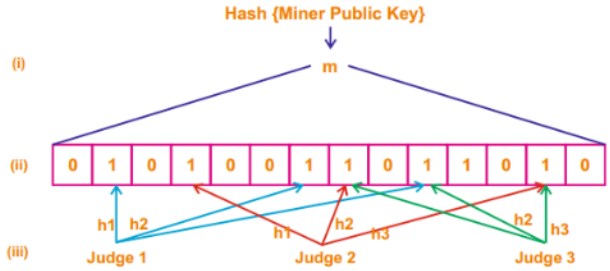


Fig. 3: Schematic representation of the BF from the hash of the miner's PK, (i) Hash algorithm on the miner's PK, (ii) First m bits resulting from the hash algorithm fill the filter, (iii) NNs check membership in the BF of the new miner

The results of the interactions represent the positions in which the filter vector must have bits in 1, if in at least one of the positions the bit is 0, the node does not belong to the set of judges of the miner of that filter. Therefore, if the respective filter positions have $1's$, the node belongs to the filter and is the miner's judge. Transactions that have filters that do not follow the bit randomness criterion, with a probability around 0.5 between $0's$ and $1's$, will be considered invalid and discarded from the network, i.e., when the number of $0's$ is much greater than the number of $1's$ in the m vector. The expected value of judges that the MiN receives according to the size of the network is calculated by the expected value of elements that are contained in the filter, according to the probability that the bits, in the k positions defined by the HFs, are 1. In Eqn. 1, p is the probability that a bit remains 0 after k iterations and n insertions. Therefore, $(1-p)$ is the probability that a bit remains 1 after k iterations and n insertions.

When checking whether an element belongs to the filter, the k positions defined by the HFs must be filled with $1's$ in the filter's bit vector. Then, the probability that an element belongs to the filter is equal to $(1-p)^k$, since filling each bit in the array is an independent process. Therefore, to calculate the expected value of judges that a node has in the network, $\hat{J}_E$, , for N nodes,

$$|E(J_E)| = \hat{J}_E = \sum_{i=1}^{N}(1-p)^k = N(1-p)^k \quad (3)$$

As the filter is implicitly formed by HF of probability $p = 0.5$, the measured value of J_E decreased exponentially, as the number of HFs, k, increases, as shown in Fig. 4(b), in a simulation in that $N = 1000$. Fig. 4(a) presents the result of the simulation of the BF test, for networks with 100 to 1000 nodes, for the selection of judges per MiN. The actual judges' values are, on average, the expected J_E judges value with 95% confidence interval, with values of at most 0.15 around the mean.

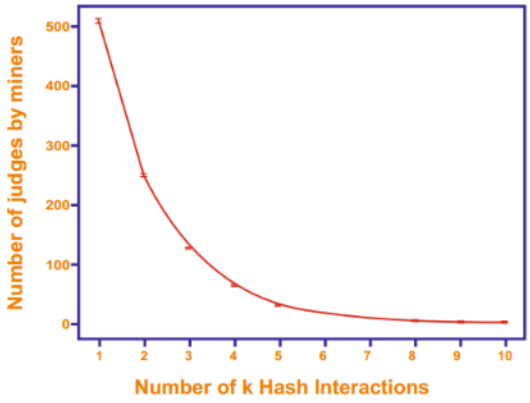


Fig. 4(a): Expected result of judges when increasing the number of k interactions in a network of 1000 nodes, where p = 0.5

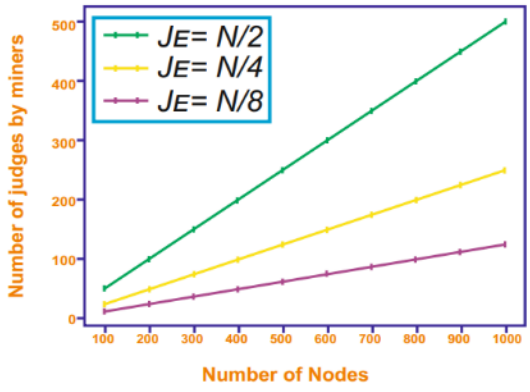


Fig. 4(b): Average NoJs is equal to the expected NoJs, according to the growth of N nodes, simulated with k = 1 for JE = N/2, k = 2 for JE = N/4 and k = 3 for JE = N/8

4.3. Monitoring of MiNs and Eviction of MNs

After the selection of judges by BF, if the node belongs to the set of J_E judges of MiN, it adds PK of the miner to the trust table and associates an Initial Reputation Score (NI). In an innocent approach, a new MiN may initially receive a high reputation score ($NI > L$). In a distrustful approach, nodes do not expect new ones to be trusted, thus receiving an initial reputation score at the trust threshold ($NI = L$). The MiN must always have a reputation score greater than or equal to the confidence threshold, $R \geq L$, to remain a network participant. Therefore, the JNs must monitor the blocks generated by their defendants and judge whether there was any malicious action. The MiNs are responsible for generating blocks both in the data blockchain and in the control blockchain. Its judges must analyze the behavior of the miner in the two chains and associate a score to each action performed [13-15]. If the miner generates a block correctly, his judges should increment the reputation score Ri = Ri-1 + Up.

If maliciousness is observed in block generation, your judges should decrement the reputation score Ri = Ri-1 - Up. R can vary between 0 and 10 and Up is an update value that can vary according to the impact of the action performed on the chain. The mechanism for expelling a MN act in two steps: voting and expulsion. In voting stage, each JN of a miner will have the reputation score R calculated and stored locally. The expulsion step will occur if the MiN receives JE + 1 Txvoting in the voting step. In this case, the MN is expelled. When updating the miner reputation scores during voting, if the judges observe that R L, where L is the confidence threshold value, they will issue the Vote transaction for expulsion, as shown in Fig. 6(a). The transaction contains PK of the possible malicious miner node, the judge's PK, the timestamp and the judge's signature on the entire content of the transaction. In order for the vote to become expulsion of MN, at least JE +1 of voting transactions must be issued. Fig. 5, in (i) the JNs of the malicious defendant issue the transaction Txvotacao. In (ii) the transactions are validated and after mining, the votes for the expulsion of the node are counted. In eviction, after mining the transactions, it is possible to determine which was the eviction voting transaction (Txvotacao) with the oldest timestamp.

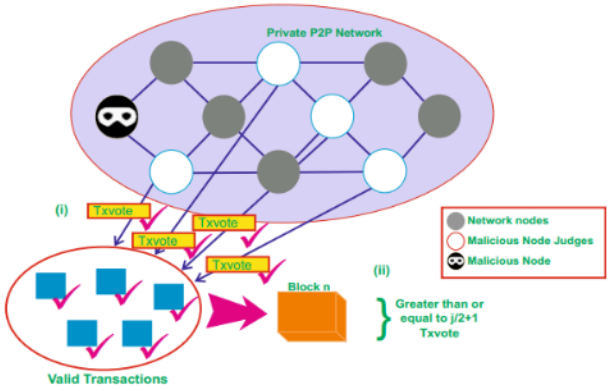


Fig. 5: Vote to expel MN schemes. Since the reputation of the judging node is lower than the upper limit of trust, the judging node sends a Txvotation transaction. Eviction vote ends when JE +1 Txvote is issued

After the voting time, the node that generated the oldest Txvotation, issues the Txexpulsion, if it does not generate the transaction, the node that generated the second oldest assumes the generation of the expulsion transaction. Fig. 6(b) shows the transaction Txexpulsion which contains PK of MN, the PK of the issuing judge of the transaction, the timestamp and the JE + 1 signatures collected in the vote for expulsion and the signature of the issuing judge on the entire transaction content. In Fig. 7, step (i), one of the judges issues the transaction Txexpulsion, which must be validated. After mining in (ii), all NNs are informed of the expulsion of the MN. In (iii), the nodes delete the PK of the expelled node. Once mined, the entire network deletes the MN key. The entire expulsion mechanism can be validated by verifying the real relevance of the JNs to the miner's BF. Expulsion transactions are only mined in a block after validating the judges' signatures. After mining, all nodes update the network view in relation to the NoN, miners or not and remove the evicted node's address, discarding its PK.

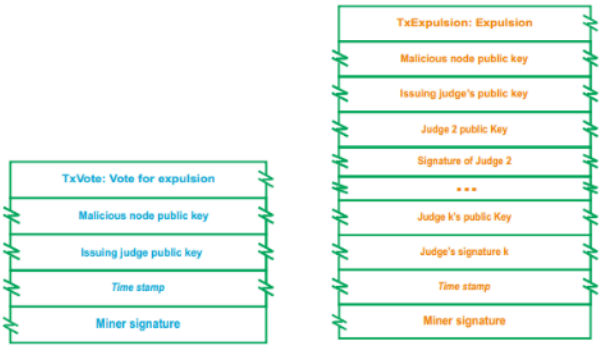


Fig. 6: MN exclusion mechanism for transactions

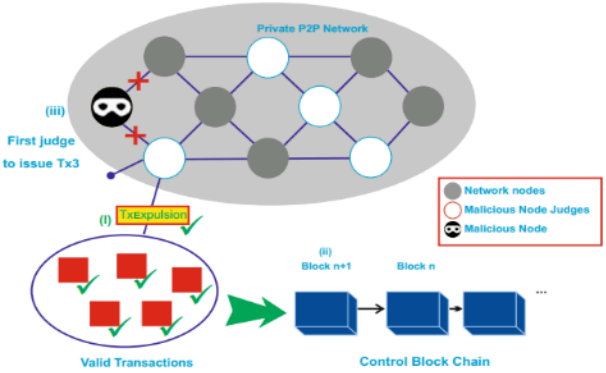


Fig. 7: Eliminate MNs. When JE+1 Txvotation votes are accumulated, the judgment node that issued the first voting transaction will send a pop-up Expulsion transaction

5. Evaluation of the trust-based CM

The evaluation of the proposed trust-based CM is carried out through simulations. For this purpose, a discrete event simulator was developed for private permissioned blockchain networks. The simulator was written in Python and all cryptography functions are implemented with PyCrypto library. The multichain platform allows an admin user to provide permissions to new NN, such as read, write and mine. Multichain's CM is PoA [28], in which nodes with mining authority take turns mining blocks. Thus, multichain is used as a basis for simulator development, as it presents good results in the performance evaluation between platforms in Chapter 3 and presents a cheap CM, but which does not assess the confidence of MiNs.

5.1. Attacker model and validation of the simulator

The evaluation of an attacker model in the context of a trust-based CM involves assessing the security and robustness of the mechanism against various types of attackers. The goal is to understand the vulnerabilities and potential attack vectors that can compromise the trust and consensus within the system. The key aspects to consider when evaluating an attacker model for a trust-based CM: adversary capabilities, attack types, trust assumptions, security guarantees, evaluation techniques and real-world considerations. Adversary capabilities determine the capabilities of the potential attackers, such as their computing power, network connectivity, knowledge of the system and access to resources. This helps in understanding the range of attacks that CM should be able to withstand. Attack types identify the different types of attacks that can be launched by adversaries. This can include double-spending attacks, sybil attacks, 51% attacks, collusion attacks and others. Each attack type represents a specific threat to the CM's integrity and trust. Trust assumptions assess the trust assumptions made by the CM. This involves evaluating how the mechanism establishes and maintains trust among participants and what assumptions it relies upon. Consider the potential vulnerabilities and weaknesses in the trust establishment process. Security guarantees evaluate the security guarantees provided by CM. Determine if it offers properties such as byzantine fault tolerance, resistance to censorship, consistency, liveness and fairness. Analyze how the mechanism mitigates attacks and ensures the integrity of the consensus. Evaluation techniques employ simulation, modeling, or formal analysis techniques to evaluate the attacker model's impact on the CM. This can involve running attack scenarios, testing the system's response and assessing its resilience.

Consider the performance, scalability and efficiency of the mechanism under attack. Real-world considerations assess how the attacker model aligns with real-world threats and scenarios. Consider the practicality of the attacks, the likelihood of their occurrence and the potential impact on the system and its users. By thoroughly evaluating the attacker model, you can gain insights into the vulnerabilities of a trust-based CM and make informed decisions about its design, implementation and security measures. The simulator is based on the behavior observed in the multichain platform for the metrics of validation time of a transaction and mining of a block, simulating the real implementation of the multichain network discrete-time simulation at each execution step representing 1 ms of actual execution. Thus, it is possible to compare the multichain execution time with the discrete simulation time estimate. The simulated p-p network is composed of 10 nodes, to simulate the network implemented on the multichain platform, all with permission to generate transactions, participate in consensus and mine blocks. Then, to validate the developed simulator, the same distribution of the time between arrival of transactions observed in the bitcoin network in the period between June 2017 and June 2018.

The ten NNs wait a time within the generalized normal Probability Distribution Function (PDF) with the parameters $\mu = 0.371$, $\alpha = 0.143$ and $\beta = 2.786$, to issue a new transaction, which varies probabilistically between 100 and 650 ms, with an average of 370 ms. The results show that the time required to mine a transaction across multiple chains and simulations follows the same PDF, Johnson's Sb, as shown in Figs. 8(a) and (b). The parameters of the two distributions differ due to inaccurate time measurements in multi-chain implementations and synchronization when using NTP servers (which implies negative times in Fig. 8(a). The discrete-time simulation does not show this inaccuracy, showing a higher concentration of histogram occurrences around the average PDF of 1.474 s. This behavior cannot be observed, as Johnson's PDF Sb of the multichain was smoothed to encompass the negative values resulting from imprecision, which justifies the difference in the parameters of the FPDs and validates the simulator.

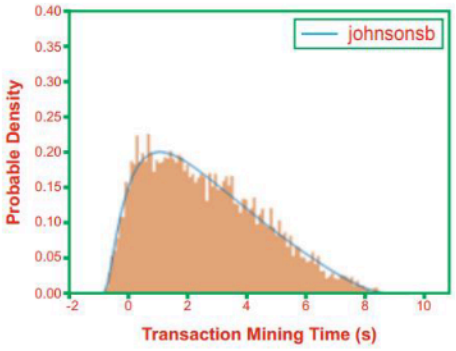


Fig. 8(a): Transaction mining time in multi-chain

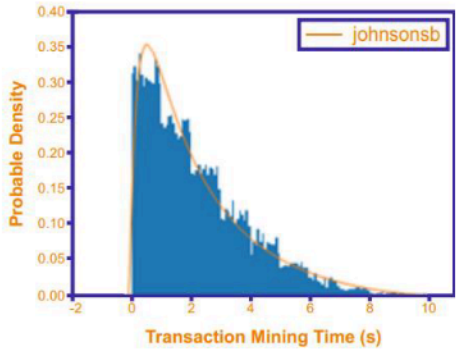


Fig. 8(b): Transaction mining time in the simulator

5.2. Simulation of access control

The simulation of the proposed access control mechanism uses the same validated simulator. Initially, with a simulated network of ten nodes, which receives ticket request transaction (Txingress) according to a uniform distribution in an interval between 1 ms and 1 h. The arrival of transactions simulates the interest of new users who want p-p network access. For the issuing user of the Txingress transaction to have access to the network, the transaction must be signed by at least one MiN. Therefore, the miner's signature represents Txingress's validation. Once validated, Txingress are mined in blocks and form an effective part of the control chain. Thus, all NNs update their view of the NoN in network N, which may trigger the need to update the number of miners (M), where M is a portion of the N nodes, which grows linearly according to the number of miners (M) with the growth of the network. If there is an opportunity for a new MiN, the node that has the longest time in the network, according to the ingress timestamp, will issue the request transaction to be a miner (Txminer), which is validated and mined following the same metrics time of the simulated network. With the objective of evaluating the interference of the number of MiNs (M) in the efficiency of the network, three network scenarios are set up, in which M varies between M = N/2, M = N/3 and M = N/4. Network scenarios, initially with N = 10, are simulated for 1 hour and the evaluation metrics are the average time (T_{avg}) for a new node to join, the T_{avg} for a node to become a miner and the loads in number of transactions and bytes generated by the access control mechanism.

The entry time metric of a new node is measured by the difference between the timestamp of the block that stores Txingress and the timestamp of emission of Txingress. Fig. 9(a) shows the T_{avg} for a user to become a NN, for the three simulated scenarios. With the analysis of the results, the M = N scenario presents the shortest probabilistic time for the access of a new node, which varies between 1 and 6 s. While in the network scenario with M = N/3 and M = N/4 the access time varies between 1 and 8 s. As much as the difference between the results of ingress time for a network with 100 nodes is subtle, it is understood that the selection of the value of M has a greater impact on access control with the growth of the network. The time metric for a node to become a miner node is measured by the difference between the timestamp of the block that stores Txminer and the timestamp of Txminer emission. Fig. 9(b) shows the T_{avg} for a node to become a miner, for the three simulated scenarios. From the analysis of the results, it is observed that the time difference is less than 1% between the three scenarios. This can be justified, as the number of Txminers is lower than the number of Txingress in the valid transaction repositories, converging the results to values similar to those measured for the entry time.

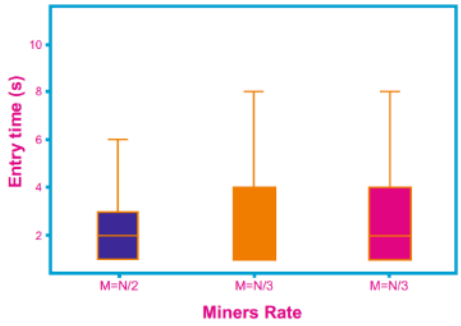


Fig. 9(a): Time from issuance to mining of Txingress

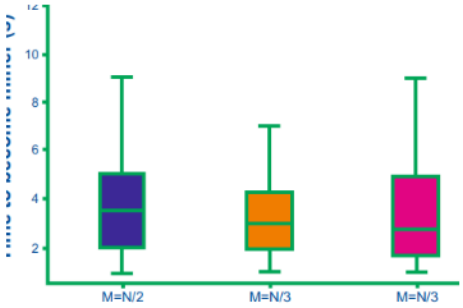


Fig. 9(b): Time from issuance to mining of Txminer

The last scenario evaluation metric is the load generated by the access control mechanism, in number of transactions and in bytes, stored in the chain of control blocks. The results are represented in Fig. 10. The loads in number of transactions are shown in Fig. 10(a) and in the number of control bytes generated in Fig. 10(b). The results show the loads generated by Txingress, which happen equally in the three scenarios and the loads generated in each of the scenarios for the Txminer transactions. The growth of miner transactions is related to the desired number of miners on the network. Therefore, the growth of the load generated by Txminer is linear to the growth of the network, following the proportion of the expected number of miners. Therefore, in one hour of simulation, the chain of control blocks accumulates 27, 434 × 103 bytes, in the M = N/2 network, while in the M = N/3 network it accumulates

18, 932 × 103 bytes and in the network of M = N/4 accumulates 14, 250 × 103. With this, load in bytes generated in the scenario from M = N/2 is twice the load at M = N/4 and approximately 1.5 times higher in M = N/3. Considering the results achieved in terms of entry time, the scenario that has Network size with miner permission M = N/2 has the shortest times. It is speed in relation to the other scenarios is justified by the greater division in the mining and validation tasks of Txingress and Txminer with more MiNs. However, for the accumulation of loads with transactions used to control access to the network, it is observed that M = N/4 gets the smallest accumulation of bytes in the string. Therefore, aiming to a network architecture that shows greater flexibility and intermediate results for the two metrics, the M = N/3 scenario, which proves to be a compromise solution between time and generated load, will be used for the next simulations.

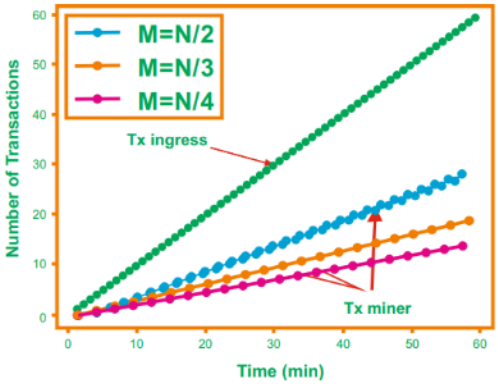


Fig. 10(a): Linear growth of number of transactions

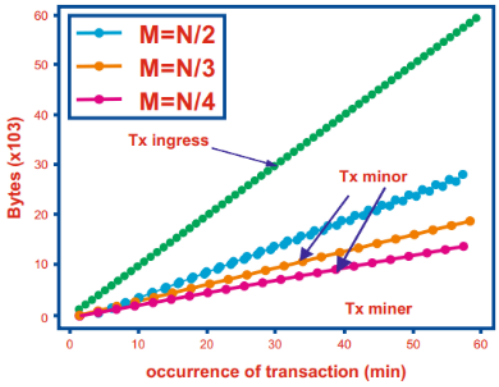


Fig. 10(b): Linear growth of the load in bytes

5.3. Monitoring and banning malicious miners

In this simulation, the network starts with N = 100 participating nodes, with the number of miners M = N/3 of the total NN, resulting in a total of M = 33 nodes miners. Within the group of MiNs, Mmalicious = M are selected to perform MAs, totaling Mmalicious = 11. The MNs have the opportunity to act at each mining round and in the simulation a probability of 0.8 is assigned to perform a MA every round. Parameters such as confidence threshold (L), initial reputation score (NI) and expected NoJs (JE) must be chosen, so that the network can detect MNs and expel them more efficiently. Therefore, 16 network scenarios are simulated, alternating parameters L, NI and JE in order to identify which configuration provides better results. All scenarios are simulated 30 times, lasting one hour, in discrete time. The No Js varies in cases where all NNs are judges of a miner (JE = N), half of the NNs are judges of a miner (JE = N/2), a quarter of the NNs are judges of a miner (JE = N/4) and an eighth of the NNs are judges of a miner (JE = N/8). For each of the judges' configurations, L and NI are configured, as shown in Table 1. The evaluation of the parameters is made from the metrics of malicious behavior time, voting time for the expulsion of a MN and load generated in bytes.

Table 1: Trust threshold values and initial reputation scores used in simulated network scenarios

	Scenarios			
L	6	7	8	6
NI	8	9	10	6

Fig. 11 presents the mean and confidence interval of time in which a MN acted on the network, from its first MA to its expulsion. Regarding the threshold scores and initial score, the scenario in which the MiN received the initial trust score equal to the threshold score is the one with the shortest time of MAs. In this scenario, the permanence of a MN is three times smaller compared to the other scenarios, while the other scenarios do not present a significant difference in time. However, when changing the scenarios in number of JE, the results show no difference in the time of perception of MNs and expulsion of these nodes. In the analysis of the voting time for eviction of a MN, Fig. 12 presents the mean result and confidence interval for the 16 network scenarios. This time is calculated from the moment the first judge issues the Txvotation, requesting the expulsion of the node until the issuance of the Txexpulsion.

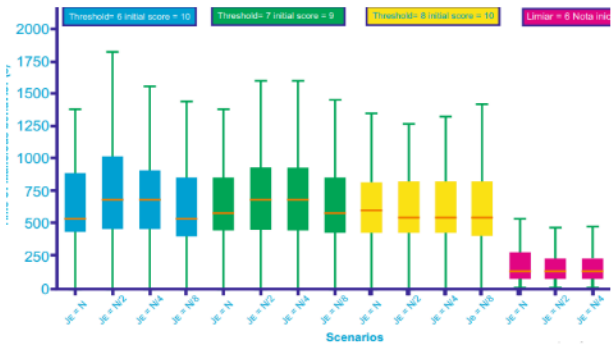


Fig. 11: Time of malicious behavior on the network, from the first action to expulsion from the network

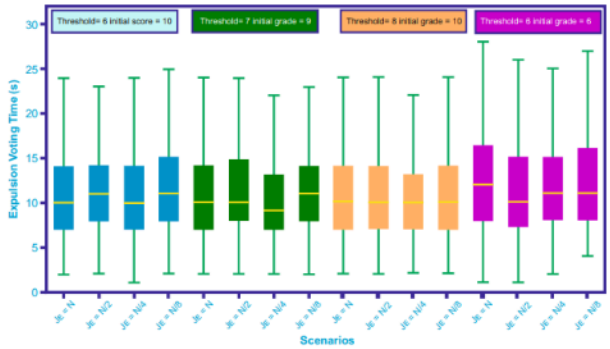


Fig. 12: Voting time for expulsion of the MN, from the issuance of the first Txvote until the issuance of the Txexpulsion transaction

The presented results show that the average voting time in all scenarios is around 10 s. It is worth mentioning that the vote converts to the expulsion of the node when (J/2)+1 Txvotacao transactions are mined and the block generation time in the simulator varies between 1 and 10 s. Therefore, the results show that Txvotacao transactions are generally mined on the same block in the chain and result in the convergence of time averages in all simulated scenarios. The confidence interval shows that some votes for eviction resulted in the total required transactions being mined in different blocks. Regarding the load in bytes accumulated with the voting and eviction transactions, Fig. 13(a) shows the linear growth of the load, according to the number of JE, in a simulation in which it is possible to expel the 11 malicious MiNs within one hour of simulation. The results show that with JE = N the load generated is two times greater than with JE = N/8, five times greater than with JE = N/4 and ten times greater that with JE = N/2, according to the proportion of transactions and signatures, which increases with the NoJs. Fig. 13(b) shows the results, on average, for the payload in bytes for thirty simulations of each proposed scenario. The larger the confidence interval and the mean value of bytes less than the measured value of successful eviction simulations, the fewer evictions occur.

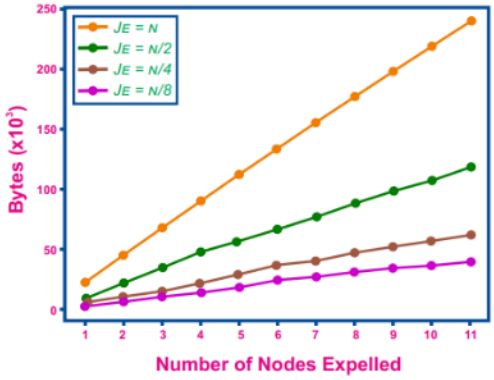


Fig. 13(a): Linear growth of payload in bytes per evictions

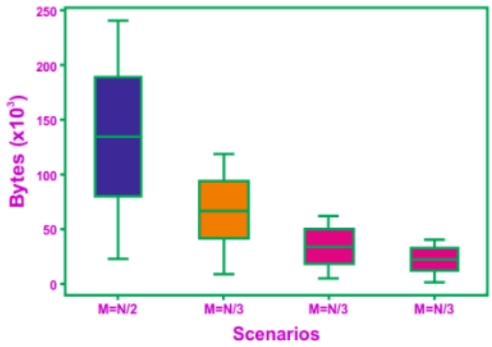


Fig. 13(b): Average payload in bytes per evictions

6. Simulation changing the proportion of MNs and MAs

From the results presented in previous simulations, it is observed that, in the scenario that adopted the initial reputation score equal to the confidence threshold (NI = L), the time in which MNs perform actions on the network is, on average, four times smaller than in the other approaches. Therefore, this approach is adopted, as it is the safest for the mechanism. Regarding the NoJs per MiN, in the scenario where JE = N is expected, the load is very high and the results are similar to scenarios with fewer judges, showing that it is not necessary to use a consensus with all the NNs on the eviction of MNs. However, the results achieved for JE = N/8 do not show the real expulsion time of all MNs, because, in some simulations, the MNs do not receive a sufficient NoJs for the vote for expulsion to take place. Therefore, the scenarios JE = N/2 and JE = N/4 are used to simulate the MA time and the proportion of MNs expelled from the network, when changing the proportion of MNs in the network and the proportion of MAs that each node performs. The proportion of MNs is varied between 10%, 20%, 30%, 40% and 50% of the network, being randomly distributed among the N = 100 nodes. Since the number of MiNs is M = N/3. Therefore, the proportion of malicious MiNs can be different in each simulation. With this, the probability of eviction of the nodes is observed, based on the total number of malicious MiNs. Fig. 14 shows the proportion of nodes kicked out in relation to the proportion of MAs on the network. This means that, in this simulation, each MN, instead of mining a block, chooses an action with probability 0, 2, 0, 4, 0, 6, 0, 8 and 1 of being malicious. For each of the proportions of MNs in the network, the probabilities of MAs are tested. Fig. 14(a) shows the average expulsion results for networks with JE = N/4 and Fig. 14(b) for networks with JE = N/2 in one-hour simulations. Despite the two networks present almost 100% eviction, when the nodes act with a probability of MAs greater than 0.6, the network with JE = N/2 has a higher proportion of expulsion even with probabilities of MAs lower than 0.6.

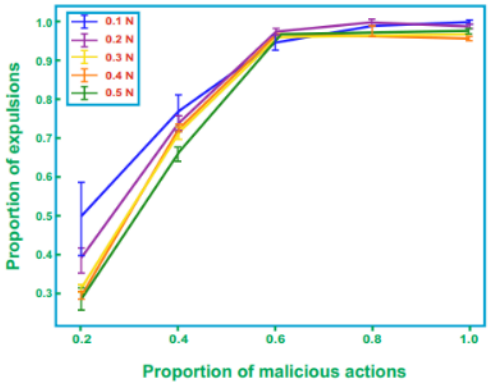


Fig. 14(a): Displays the eviction proportion for networks, JE = 25

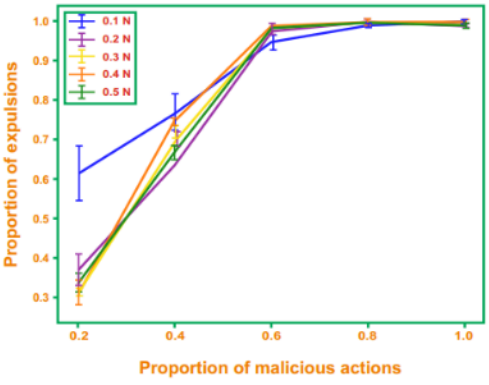
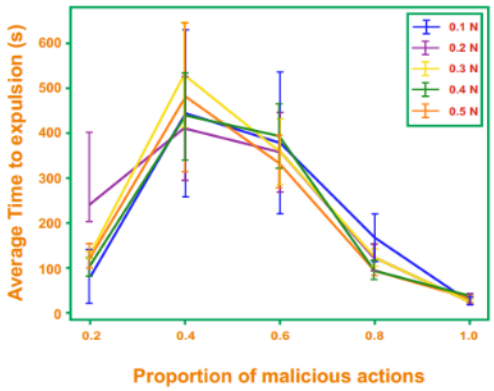
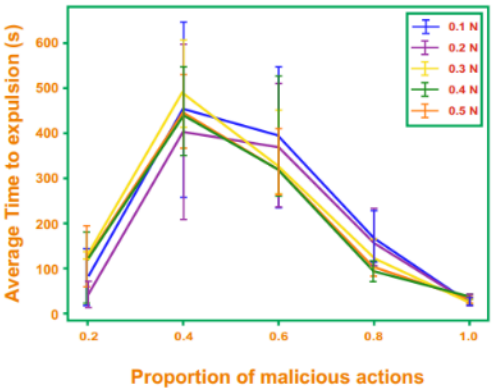


Fig. 14(b): Displays the eviction proportion for networks, JE = 50

The eviction time of MNs in the network is the interval from when a MN performs the first MA until its expulsion. Fig. 15 shows the Tavg for each proportion of malicious NNs and their probability for MAs. Fig. 15(a) presents the Tavg for networks with JE = N/4 and Fig. 15(b) for networks with JE = N/2. The results do not show much difference between the two networks. It should be noted that when the probability of malicious behavior is 0.2, the time consequences of malicious behavior are low. Because it is impossible to exclude all MNs in this scenario in a 1-hour simulation and with probability 0.2. It is possible that none of them will be executed. These results represent only those nodes that behaved maliciously and were excluded during the simulation. Therefore, there may be nodes with longer periods of malicious behavior, but they are not included in this analysis. In the calculation results, about 70% of the nodes that were kicked out of the network with a malicious behavior probability of 0.4 are the nodes with the longest behavior period. The results are similar for both scenarios. Observing the results, when the probability of MA is 0.2, the time to expel the MNs is smaller, as it is not possible to expel all the MNs in one hour of simulation. So, as the two scenarios present similar results, even changing the proportions of MNs and the probabilities of MAs, the best approach is JE = N/4, as it accumulates less expulsion transaction loads.



(a) Mean eviction time of MNs for JE network = N



(b) Mean eviction time of MNs for JE network = N

Fig. 15: Time to the expulsion of the node from the network during the first MA, changing the probability of acting maliciously

6.1. Hybrid PKI and blockchain system

A hybrid PKI (Public Key Infrastructure) and blockchain system combines the strengths of both technologies to enhance security, privacy and trust in various applications. PKI is a system that enables the issuance, distribution and management of digital certificates used for authentication, encryption and digital signatures. In a traditional PKI system, a central certificate authority (CA) is responsible for issuing and verifying digital certificates. CAs play a crucial role in establishing trust and validating the authenticity of PKs. And blockchain is a decentralized and immutable distributed ledger that provides transparency, security and tamper-resistance. It achieves consensus among participants through cryptographic mechanisms, such as PoW, ensuring that the data recorded on the blockchain is trustworthy and cannot be easily altered. Integrating PKI with blockchain can leverage the advantages of both technologies. The PKI component is responsible for issuing and managing digital certificates. The CA can issue digital certificates to entities (users, devices, or organizations) and bind their PKs to their identities.

The blockchain component stores a decentralized ledger of all issued certificates, their status and associated metadata. Validators within the blockchain network can verify the validity of certificates by accessing the blockchain's distributed ledger, eliminating the reliance on a central authority. If a certificate needs to be revoked due to compromise or expiration, the revocation status can be recorded on the blockchain, ensuring the information is globally accessible. The blockchain component establishes a decentralized network where participants reach consensus on the state of the blockchain. By leveraging consensus algorithms, such as PoW or PoS, the blockchain ensures that only valid and authorized transactions, including certificate issuance and revocation, are accepted and recorded. Combining PKI with blockchain enhances security and transparency in the system. The immutability of the blockchain prevents unauthorized modifications to certificate-related data, providing a tamper-resistant and auditable record. The decentralized nature of the blockchain removes single points of failure and reduces the risk of a single compromised entity compromising the entire system.

6.2. Simulation results

The main idea of the simulation is to evaluate the real average write time of EMR in the blockchain and the overhead of writing EMR records. The time to write an EMR to the chain is the difference between when the EMR was created and when the block containing the EMR transaction was mined. Therefore, the first experiment evaluates transaction recovery times for different network sizes. Scenarios with 100 to 1000 nodes are simulated, as shown in Fig. 16(a). The results show that the Tavg to write a transaction in the chain remains between 2 and 3 seconds in all the evaluation scenarios. The effect of the number of MiNs on network performance is also evaluated. Although the number of MiNs responsible for bringing blocks into the chain has increased, it can be seen in Fig. 16(b) that the transaction mining time remains the same when each MiN produces a block. Therefore, increasing the number of MiNs reduces the load on each node, but does not affect the time it takes for the EMR to enter the blockchain. The load generated in one-hour simulation was also

evaluated. Fig. 16(c) shows that the chain size increases linearly as the simulation progresses, indicating that the chain begins to grow after 8 minutes. This is because the simulation parameters take into account that the interval between transactions is greater than or equal to 8 minutes. Also, as the NoN in the network increases, the size of the chain increases as more nodes generate transactions. Therefore, the results presented in Fig. 16 support the idea that the proposed blockchain-based EMR scaling method is proportional to NoN in the network and guarantees data availability.

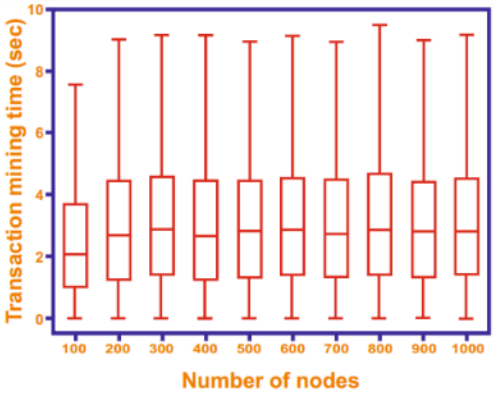


Fig. 16(a): Transaction execution time remains constant as the network grows, indicating that the offering is scalable

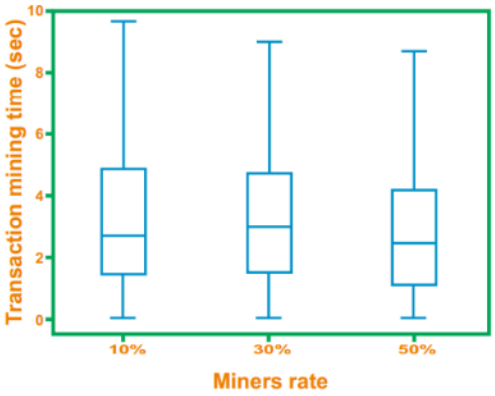


Fig. 16(b): Number of miners on the network does not affect the execution time of transactions

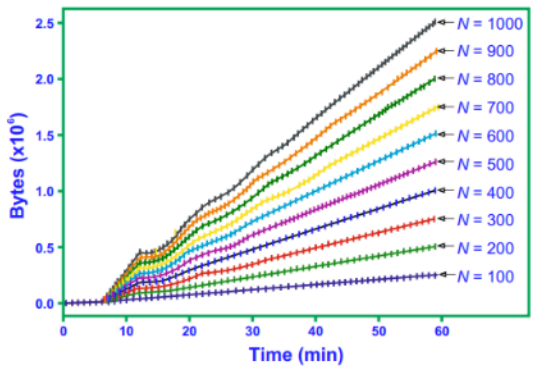


Fig. 16(c): More transaction-generating nodes in the network, the higher the storage overhead

7. Conclusion

It is concluded that the proposed CM is efficient for the self-organization of the network in a distributed manner. In addition, the mechanism offers constant monitoring of the MiNs, based on a pseudo-random jury, unlikely to be pre-selected by the monitored MiN since if manipulation is observed in key generation, the node that is handling key generation is already eliminated in the access request phase. The mechanism also proved to be efficient for the expulsion of MNs that acted with more than 50. From the proposed CM, a hybrid approach was developed that combines the advantages of the blockchain. A PKI, to develop a system for electronic health records, capable of providing security and privacy, with patient-centered access control. Although the use of the PKI is mandatory in some countries, the proposal complies with the requirements required by Indian law for electronic medical records and is truly distributed, with a high level of privacy for the patient's health records. This approach also maintains confidentiality between patient and physician due to the validation of all transactions that are added to the chain, since disclosing a patient key between physicians is prohibited. Furthermore, the results obtained through simulation show that the approach is scalable, as the time to mine transactions remains constant when the NoN in the network increases and the chain size increases linearly with the growth of the network.

REFERENCES:

[1] H. Hu, K. Xiong, G. Qu, Q. Ni, P. Fan and K.B. Letaief. 2021. AoI-minimal trajectory planning and data collection in UAV-assisted wireless powered IoT networks, *IEEE Internet of Things J.*, 8(2), 1211-1223. https://doi.org/10.1109/JIOT.2020.3012835.

[2] R. Vaishnavi, J. Anand and R. Janarthanan. 2009. Efficient Security for desktop data grid using cryptographic protocol, *Proc. IEEE Int. Conf. on Control, Automation, Communication and Energy Conservation*, Perundurai, India.

[3] J.P. Qian, Q.Y. Yu, S.Y. Shi, B. Zhang, Y. Zha and W. Wu. 2020. Design of an intelligent control platform for agricultural inputs based on blockchain, *J. Agricultural Big Data*, 2(2), 38-46.

[4] H.L. Yu, B.Y. Chen, D.M. Xu, X. Yang and C. Sun. 2020. Modeling of rice supply chain traceability information protection based on block chain, *Trans. Chinese Society for Agricultural Machinery*, 51(8), 328-335.

[5] J. Anand, C. Aasish, S.S. Narayanan and R.A. Ahmed. 2023. *Drones for Disaster Response & Mgmnt.*, 1st Ed., CRC Press. https://doi.org/10.1201/9781003252085-11.

[6] R. Sparrow, M. Howard and C. Degeling. 2021. Managing the risks of artificial intelligence in agriculture, *NJAS: Impact in Agricultural and Life Sci.*, 93(1), 172-196. https://doi.org/10.1080/27685241.2021.2008777.

[7] L. Wang, Y. He and Z. Wu. 2022. Design of a blockchain-enabled traceability system framework for food supply chains, *Foods*, 11, 744. https://doi.org/10.3390/foods11050744.

[8] S.M. Alrubei, E.A. Ball, J.M. Rigelsford and C.A. Willis. 2020. Latency and performance analyses of real-world wireless IoT-blockchain application, *IEEE Sensors J.*, 20(13), 7372-7383. https://doi.org/10.1109/JSEN.2020.2979031.

[9] A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher and F. Wang. 2017. Secure and trustable electronic medical records sharing using blockchain, *AMIA Annual Symp. Proc.*, 17, 650-659.

[10] T.T.A. Dinh, J. Wang, G. Chen, R. Liu, B.C. Ooi and K.L. Tan. 2017. BLOCKBENCH: A framework for analyzing private blockchains, *Proc. 2017 ACM Int. Conf. on Management of Data*, 1085-1100.

[11] M. Castro and B. Liskov. 1999. Practical byzantine fault tolerance, *Symp. Operating Systems Design and Implementation*, 173-186.

[12] M. Pilkington. 2016. *Blockchain Tech.: Principles and Applications*, Edward Elgar Pub., UK. https://doi.org/10.4337/9781784717766.00019.

[13] S. Gupta and M. Sadoghi. 2018. *Blockchain Transaction Process.*, 1-21. https://doi.org/10.1007/978-3-319-63962-8_333-1.

[14] K. Christidis and M. Devetsikiotis. 2016. Blockchains and smart contracts for the internet of things, *IEEE Access*, 4, 2292-2303. https://doi.org/10.1109/ACCESS.2016.2566339.

[15] R. Dhanalakshmi, J. Anand, K. Poonkavithai and V. Vijayakumar. 2022. Cloud-based glaucoma diagnosis in medical imaging using machine learning, *Artificial Intelligence for Innovative Healthcare Informatics*, 61-78. https://doi.org/10.1007/978-3-030-96569-3_3.

[16] G. Rekha, K. Gulshan and A. Mamoun 2021. A secure localization scheme based on trust assessment for WSNs using blockchain technology, *Future Generation Computer Systems*, 125(11), 221-231. https://doi.org/10.1016/j.future.2021.06.039.

[17] L.H.G. Ferraz, P.B. Velloso and O.C.M. Duarte. 2014. An accurate and precise malicious node exclusion mechanism for ad hoc networks, *Ad hoc Networks*, 19, 142-155. https://doi.org/10.1016/j.adhoc.2014.03.001.

[18] P.B. Velloso, R.P. Laufer, O.C. Duarte and G. Pujolle. 2006. A human- inspired trust model, *Mobile and Wireless Communication Networks*, 35-46. https://doi.org/10.1007/978-0-387-34736-3_2.

[19] M. Virendra, M. Jadliwala, M. Chandrasekaran and S. Upadhyaya. 2005. Quan- tifying trust in mobile ad-hoc networks, *Integration of Knowledge Intensive Multi Agent Systems*, 65-70.

[20] S. Zhu, S. Xu, S. Setia and S. Jajodia. 2003. Lhap: A lightweight hop-by-hop authentication protocol for ad-hoc networks, *Distributed Computing Systems Workshops*, 749-755.

[21] D. Schwartz, N. Youngs and A. Britto. 2014. The ripple protocol consensus algorithm, *Ripple Labs Inc*., 1-8.

[22] D. Ongaro and J. Ousterhout. 2014. In search of an understandable consensus algorithm, *Proc. USENIX Conf. on USENIX Annual Tech. Conf*., Philadelphia, US.

[23] L. Lamport. 2001. Paxos made simple, *ACM SIGACT News*, 32(4), 18-25.

[24] M.R. Islam, M.M. Rahman, M. Mahmud, M.A. Rahman and M.H.S. Mohamad. 2021. A review on blockchain security issues and challenges, *Proc. IEEE 12th Control and System Graduate Research Colloquium*, Shah Alam, Malaysia. https://doi.org/10.1109/ICSGRC53186.2021.9515276.

[25] N. Sharma, M. Shamkuwar and I. Singh. 2019. The history, present and future with IoT, *Internet of Things and Big Data Analytics for Smart Generation*, Springer, Berlin/Heidelberg, Germany. https://doi.org/10.1007/978-3-030-04203-5_3.

[26] S. Zhu, S. Xu, S. Setia and S. Jajodia. 2003. A lightweight hop-by-hop authentication protocol for ad-hoc networks, *Proc. 23rd Int. Conf on Distributed computing Systems Workshops*, Providence, RI, USA.

[27] R. Thillaikkarasi, M.M. Yaseen, R. Rameshbabu, R. Prabhakaran, R. Kesavan and A.J. Anand. 2023. Waysides inspection using wayside processing imaging and deep learning, *3rd Int. Conf. Pervasive Computing and Social Networking*, Salem, India. https://doi.org/10.1109/ICPCSN58827.2023.00065.

[28] M. Anwar, F. Masud, R.A. Butt, S.M. Idrus, M.N. Ahmad and M.Y. Bajuri. Traffic priority-aware medical data dissemination scheme for IoT based WBASN healthcare applications, *Computers, Materials & Continua*, 71(3), 4443-4456. https://doi.org/10.32604/cmc.2022.022826.

[29] J. Anand, T.G.A. Flora and A.S. Philip. 2013. Finger-vein based biometric security system, *Int. J. Research in Engg. and Tech*., 2(12), 197-200. https://doi.org/10.15623/ijret.2013.0212035.

[30] S. Chung, J.T. Hwang and S.H. Park. 2022. Physiological effects of bioactive compounds derived from whole grains on cardiovascular and metabolic diseases, Applied Sci., 12(2), 658. https://doi.org/10.3390/app12020658.

[31] H.M. Moyeenudin, M. Narender, J. Amutharaj, S. Harikumar and A.J. Anand. Comparative analysis of video transmission in vehicular networks using IEEE 802.11g and IEEE 802.11p Standards, *Proc. Ist Int. Conf. on Advances in Electrical, Electronics and Computational Intelligenc*e, Tiruchengode, India.

[32] S. Mahalakshmi, C. Rajeswari, A.J. Anand and A. Rahul. 2022. Client authentication by signature verification method using robotic process automation (RPA), *Proc. Int. Conf. on Data Sci., Agents & Artificial Intelligence*, Chennai, India. https://doi.org/10.1109/ICDSAAI55433.2022.10028810.

[33] I.D. Alvarenga, G.A.F. Rebello and O.C.M.B. Duarte. 2018. Securing configuration management and migration of virtual network functions using blockchain, *Proc. of IEEE/IFIP Network Operations & Management Symp.*, Taipei, Taiwan. https://doi.org/10.1109/NOMS.2018.8406249.

[34] I. Eyal and E.G. Sirer. 2018. Majority is not enough: Bitcoin mining is vulnerable, *Communications of The ACM 61*, 7, 95-102. https://doi.org/10.1145/3212998.

[35] K. Qin, B. Fu, P. Chen and J. Huang. 2020. Multicost rerouting algorithm in SDN, *J. Advanced Computational Intelligence and Intelligent Informatics*, 24(6), 728-737. https://doi.org/10.20965/jaciii.2020.p0728.