IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 October 2022, accepted 7 November 2022, date of publication 17 November 2022,
date of current version 30 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3222807

RESEARCH ARTICLE

CTB-PKI: Clustering and Trust Enabled Blockchain Based PKI System for Efficient Communication in P2P Network

AMRUTANSHU PANIGRAHI[1], (Member, IEEE), AJIT KUMAR NAYAK[2], (Member, IEEE), ROURAB PAUL[1], (Member, IEEE), BIBHUPRASAD SAHU[3], (Member, IEEE), AND SHASHI KANT[4], (Member, IEEE)

[1]Department of Computer Science and Engineering, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha 751030, India
[2]Department of Computer Science and Information Technology, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha 751030, India
[3]Department of AI and DS, Vardhaman College of Engineering (Autonomous), Hyderabad, Telangana 501218, India
[4]Department of Management, College of Business and Economics, Bule Hora University, Ethiopia, Africa

Corresponding author: Shashi Kant (shashi.kant@bhu.edu.et)

ABSTRACT The decentralization feature of public and private blockchain-based applications is achieved by selecting different nodes as validators or Certificate Authority (*CA*) for each transaction. Public blockchain uses Proof of Work (*PoW*) to search for the validator. *PoW* causes an enormous amount of energy. Therefore, Proof of Stake (*PoS*), and Proof of Authority (*PoA*) emerged as alternate solutions. Selection of a new *CA* using *PoS* or *PoA* algorithms for each transaction may improve transaction security. However, a network may have a large number of transactions and participants. Selecting a *CA* for each transaction using *PoS* or *PoA* may cause a significant amount of block propagation delay, which can reduce network efficiency drastically. This paper proposes a different approach to increase the efficiency of Blockchain-based Public Key Infrastructure ($BC-PKI$). The proposed approach creates clusters of participant nodes based on their validation time, response time, and trust. This method selects a cluster based on the budget of response time and validation time given by the node that intends to start a transaction. Thereafter, the node which has the highest trust in that cluster is chosen as a *CA* for the next transaction. Instead of searching on all participant nodes, our approach searches on the nodes of the chosen cluster which reduces the searching space of the *CA* selection process. This research work adopts a trust evaluation approach where the trust factor is quantified based on its experience and reputation. The node trust is reevaluated after every successful and unsuccessful transaction. A node that performs more successful transactions has more trust value. The node that has a higher trust value has a higher probability to be selected as a *CA* for a transaction. The trust reevaluation process is followed by the clustering process. The result shows the proposed approach can reduce ∼38.5% response time and ∼2.2% validation time as compared to infrastructure which does not implement clustering. Additionally, the proposed $CTB-PKI$ can be used in Blockchain 2.0 and Blockchain 3.0-related applications.

INDEX TERMS Blockchain, certificate authority, clustering, public key infrastructure, smart contract.

I. INTRODUCTION

Communication via an unprotected network can only be guaranteed by the verification of each participant's identity. For example, a man-in-the-middle (MITM) attack [1] may be used to intercept communication and imitate the participant's involvement. Public Key cryptography [2] is one of the promising solutions to secure communication in an untrusted network. Since the introduction of public key cryptography, the verification of the trustworthiness of a participant's public key has been a prominent issue. In this context, ''trusted'' means that the private key is known only to the intended communication partner. If both participants involved in the communication know the same secret, such

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan.

as a password, the problem becomes simplified significantly. Sharing a private key in a large-scale network is not always possible. So Public Key Infrastructure (PKI) [3] can be used as an alternative solution for public key cryptography.

The use of encrypted communication protocols is being actively pushed and supported more than it has ever been before. Regular HTTP connections, which appeared to be fair in the past [4], are now described as ''not secure,'' whereas HTTPS connections are unmistakably labeled as ''secure.'' This change in the appearance of security indicators in the address bar has been implemented by browser vendors such as Google Chrome [5], Mozilla Firefox [6], and other popular browsers. Cryptographically protected protocols such as Transport Layer Security (TLS) protocol are becoming the usual solutions as more administrators and developers become aware of the risks associated with using insecure protocols [7]. The risks associated with insecure protocols are increasing as more people become aware of them. The TLS *PKI* also has a vulnerability called the weakest-link security problem, which means that any trustworthy *CA* may create a valid certificate on its own for any domain name. A client will regard a certification authority to be trustworthy if that authority's certificate is included in the client's list of root CAs or if the certificate was signed by another trusted CA. Both $X.509$ [8] and *PGP* [9] are the other two widely used protocols for securing internet-based communication.

An attacker may undermine the integrity of the system as a whole by gaining control of a single root or intermediate *CA*s. To overcome the issues caused by the generalized web-based security protocols there are two different solutions present Log-based and Web of trust. Among all solutions, *CertificateTransparency*(*CT*) framework [10] by Google is the most popular one. It makes certificates publicly accessible by using append-only logs for updating and maintaining the list of log servers. Even if the contents of the log may be read and shown to be consistent, log servers have the option to disregard any requests that are sent their way. Last but not least, a gossip protocol is required in order to prevent a split-world attack [11], which occurs when a malicious log server presents various clients with conflicting copies of the log. Therefore, in order to accept a malicious or compromised CA, each certificate issuance should include numerous *CAs*, and all activities should be documented in a safe and completely dispersed manner. This is necessary in order to tolerate the presence of a CA. The other $Log - basedPKI$ solutions include *AccountableKeyInfrastructure*(*AKI*) [12], *AttackResilienceKeyInfrastructure*(*ARPKI*) [13], etc. The main issue the log-based PKI is facing is the centralized *IntegratedLogServer*(*ILS*). The presence of ILS makes the Log-based PKI solution prone to a single-point failure. The third-party can easily get access to the ILS server by means of which the entire system will fail to maintain the integrity level. Another possible solution to the conventional PKI system is the *WebofTrust*(*WoT*) based PKI. The *WebofTrust*(*WoT*) includes notary-based solutions such as *LocalPKI* [[14] and $Notary - basedPKI$ [15] that are intended to offer different PKI systems that enable the end-user to use their known trusted node to act as the CA. In this type of PKI, the *NotaryAuthorities*(*NAs*) replaced CAs to store the only signed hash of the certificate and its serial number in the database. However, with $notary - basedPKI$ and *LocalPKI* systems, users and NAs must have confidence in order to oversee the functioning of certificates. Therefore, it is necessary to prevent notaries from certifying bogus certificates and signatures.

To overcome the lacuna present in $Log - basedPKI$ and *WoT*, blockchain-based PKI becomes an emerging solution. The characteristics such as immutability, transparency, security, and distributed ledger are the technical benefits of blockchain which make it a more appropriate technique for internet-based communication. A promising characteristic known as decentralization of internet services is the key concept presented behind blockchain technology. Instead of depending on a single *CA* for issuing the certificate, this technique enables the network to have multiple *CAs* for different communications. Adopting multiple CAs simply avoids the single-point failure limitation of conventional PKI systems.

A. PAPER STRUCTURE

The rest of this paper is structured in the following manner. Section II shows the problem statement, motivation in addition to the key contribution of the research work. The section III focuses on the existing literature based on P2P and blockchain network trust calculation (subsection III-A), blockchain-based clustering (subsection III-B), and blockchain-based PKIs (subsection III-C). Section IV focuses on the preliminary study for the proposed work. The proposed $CTB - PKI$ is reported in Section V. Section VI presents the working principle of the proposed work. The proposed PKI is evaluated based on different parameters in Section VII. Finally, Section VIII shows the conclusion of the research work.

II. PROBLEM STATEMENT AND MOTIVATION

Evidently, the decentralization characteristic eliminates the limitations inherent in the conventional centralized PKI system. In a blockchain network, every transaction requires the selection of a *CA*. Therefore, a large number of transactions need extensive computing effort. This *CA* selection procedure becomes the major cause of network computation overhead, which reduces the network's performance. To circumvent the problem, this network clustering is a potential solution.

In addition, the blockchain nodes perform the transaction with the other participant nodes with the presence of some participant node called as *CA*. In this node interaction process, trust is the key factor. The so-called ''don't trust'' issue of blockchain considers a poor relationship among all nodes. Even though BC-PKI has transparency, decentralization, immutability, and security still it faces a credibility crisis. A credibility crisis explains a scenario of whether the participant nodes are creditable or not for a successful transaction. Choosing a node as a *CA* which performs more

number of successful transactions will increase the trust of that node and the efficiency of the PKI as well. Hence trust value can be one of the most important parameters for the BC-PKI network. Therefore, the current work considers the clustering of participant nodes of the blockchain network and trust value calculation as the most inclusive factors.

The key contributions of this current research work are summarized below:

- The proposed $CTB - PKI$ implements a cluster-based CA selection approach which reduces search spaces significantly. As a result, the CA selection process can save $\sim$ 38.5% response time and $\sim$ 2.2% validation time. The clustering algorithms used in our $CTB - PKI$ CA selection process are based on 3 parameters: trust, response time, and validation time. The proposed $CTB - PKI$ uses the $K - Means$ with *silhouettescore* and *DBScan* clustering algorithms.
- The proposed $CTB - PKI$ quantified the trust value-based experience and reputation of the participant node. The reputation is based on direct and indirect trust and the experience is calculated based on the number of successful and unsuccessful past transactions
- The proposed PKI is evaluated based on the three metrics (i) response time with and without clustering, (ii)validation time with and without clustering, and (iii) Gas cost used for different transactions. The reduced latency of the proposed $CTB - PKI$ makes it suitable for Blockchain 2.0 and 3.0 application domains.

III. RELATED WORK

This work is mainly motivated by 3 major issues (i) Trust calculation, (ii)Clustering of participant nodes to reduce the searching space Validator, and finally (iii) PKIs. In section III-A, existing literature on the trust calculation of node in a Point to Point (P2P) network with and without blockchain are discussed. In the section III-B different blockchain-based clustering mechanisms are discussed where machine learning plays a crucial role. In section III-C various blockchain-based PKI systems are discussed.

A. P2P NETWORK AND BLOCKCHAIN NETWORK TRUST CALCULATION

In this section, various trust calculation methods in P2P network (section III-A and Table 1) along with the blockchain network node trust calculation methods (section III-A and Table 2) are reported.

(i) **P2P network trust calculation**: The Bayesian network trust model introduced by Wang and Vassileva [16] employs the Bayesian network to compute the trust degree and the probability technique to determine the node trust value, which subtly increases algorithmic complexity. The trust parameters are quantified into the [−1,1] range, which may be stated intuitively as a full trust to total untrust node. A model for calculating trust based on evidence theory was proposed by Yu and Singh [17]. Evidence of a node's support has been used to recognize that particular node as the target node.

A distributed trust calculation model called PageRank was suggested by Yamamoto et al. [18]. This model estimates the trust value of nodes by using the PageRank algorithm that is shared throughout the network. PeerTrust, which was developed by Xiong et al., makes use of many factors to automatically alter the trust value of nodes over time, ultimately selecting the high-trust node as the one with which to connect [19]. PeerTrust determines the trustworthiness of a node by taking into account a number of criteria relating to a transaction and the environment of the network. Based on the D-S evidence theory, Wen et al. [20] suggested a way to identify trust relationships and confidence intervals between peers. In order to determine the reliability of the nodes, the model makes use of both the arithmetic average and the Bayesian approach simultaneously.

Song et al. [21] presented a model for the trust that makes use of fuzzy logic inference to calculate the local trust value of a peer and aggregates the recommendation information. The principles for logical reasoning using linguistic trust metrics are provided by fuzzy logic. For DHT-based P2P networks, the PowerTrust system [22] was suggested, which makes use of the Power-law distribution of peer feedback. Using a distributed ranking method, PowerTrust dynamically chooses a limited number of power nodes that are the most trustworthy. PowerTrust dramatically increases global reputation accuracy and aggregation speed by using a look-ahead random walk approach with the power nodes.

(ii) **Blockchain network trust Model:** Sun et al. [23] proposed a trust calculation model for a blockchain network that calculates the trust value of a node by acquiring the working state and behavioral information of that intended node. The final trust is calculated by aggregating the trust value generated during the transaction and the trust value generated by the behavior. For a blockchain-based online payment system, Ahn et al. [24] suggested a methodology for estimating trust and reputation using the values contained on a blockchain ledger. Information from ratings and transaction histories has been effectively utilized to

TABLE 1. Related work based on P2P network trust calculation.

Ref	Trust calculation model	Network type	Blockchain Implementation
[16]	Baysian Network	Peer-to-Peer	✗
[17]	Evidence Theory	Peer-to-Peer	✗
[18]	Page Rank Algorithm	Peer-to-Peer	✗
[19]	Successful transaction	Peer-to-Peer	✗
[20]	Evidence Theory	Peer-to-Peer	✗
[21]	Fuzzy logic inference	Peer-to-Peer	✗
[22]	Node Feedback	Peer-to-Peer	✗

calculate reputation and trust levels. The blockchain-based payment system keeps track of its entire history. While regularly validating and confirming such values in the background for dependability without impairing user experience, the model uses a small cache of key data to speed up searches.

She et al. [25] proposed a blockchain-based trust model for detecting the malicious node in the case of the wireless sensor node. For calculating the trust value four different attributes including the node behavior, response time, transmission delay and forwarding rate have been considered. The state of the node is further divided into two different groups such as working or non-working state. Initially, a node has been verified for its state and if the state is found working then only the other three parameters are considered otherwise the node will be discarded from the network. The final trust has been calculated by aggregating the delay factor, forwarding rate, and response time.

Zhao et al. [26] presented a model Trustblock to calculate the trust of the data layer devices for the Software Defined Network (SDN). Direct, Indirect, and Historical trust are the three key parameters considered for calculating the final trust of a node. The final comprehensive trust is calculated by normalizing the three different types of trust with three different weight factors $w1$, $w2$, and $w3$. These weights are calculated by using the entropy value. Inedjaren et al. [27] have proposed a blockchain-based distributed framework for calculating the trust in the Vehicular Adhoc Network (VANET). The node uses two types of control messages such as *HELLO* and Traffic Control (*TC*) through the OLSR routing protocol for any kind of communication. The trust of the node is calculated by using the membership value of each control message which can be of *verylow*, *low*, *medium*, *large*, and *verylarge*. Next, the defuzzification rule is applied to the membership value of *HELLO* and *TC* to obtain the final trust value of that particular node.

TABLE 2. Related work based on Blockchain trust model.

Ref	Key Feature	Blockchain Platform	Trust Usage	
			PKI	CA Selection
[23]	Behavioral information	✓	✗	✗
[24]	Transaction History	✓	✗	✗
[25]	Node behaviour, response time, transmission delay, data forwarding rate	✓	✗	✗
[26]	Transaction history and Peer feedback	✓	✗	✗
[27]	Defuzzification	✓	✗	✗

B. BLOCKCHAIN CLUSTERING

In the current section, different machine learning-based clustering approaches for blockchain networks are discussed. Table 3 shows the summarization of the considered literatures.

Zola et al. [28] proposed a machine learning-based method for detecting malicious activities in a bitcoin network. Initially, the clustering algorithm has been applied in order to make different clusters of malicious and non-malicious data present in the blockchain. Finally, different ensemble machine learning by using different classification algorithms such as Random Forest, Adaboost, and Gradient Boosting for classification purposes. The proposed model shows a 99.68% accuracy level. Chawathe et al. [29] proposed a novel approach for clustering the bitcoin data for behavioral analysis. For clustering, the K-Means clustering algorithm has been considered. Mahalanobis distance metrics have been used in order to evaluate the identified clusters.

Huang et al. [30] proposed a novel approach as the Behavior Pattern Clustering (BPA) algorithm which takes the blockchain transactional data over time as the input. The proposed algorithm has been evaluated by considering 1321 numbers of records as the nodes. BPA has been compared with the K-Means and K-Means ++ algorithms to show its efficiency. Ermilov et al. [31] reported an approach to identify the blockchain data owner. This can e performed by using behavioral pattern analysis and off-chain data available publicly. For behavioral pattern analysis, machine learning clustering algorithms have been applied. Web crawling and manual analysis of various bitcoin data providers are used for the off-chain data analysis.

Harrigan and Fretter [32] have used the machine learning clustering algorithm for making different clusters of the available bitcoin data available up to February 2016. The clustering method has been implemented to the publicly available data to identify fraudulent transactions. As a result, the author has created a supercluster of the identified attacks. Fleder et al. [33] reported a novel approach by using the machine learning clustering algorithm to identify the known and unknown users. For empirical analysis, the raw bitcoin data up to December 2013 has been considered.

C. BLOCKCHAIN BASED PKI

Garba et al. [34] proposed a blockchain-based PKI $BB-PKI$ for managing the certificates. In this, a client initially requests a certificate from the registering authority (RA), and then the RA forwards the request message to the corresponding CA for certificate issuance. Within the network, there are multiple CAs and RAs. The main objective of this work is to avoid the single point of failure (SPoF). Lukasz et al. [35] proposed a blockchain-based PKI known as *BlockPKI*. The main objective of this model is to automate the certificate issuance system. The domain owner defines the number of CAs who can issue and validate the certificate. Upon receiving the request from the node for a certificate the smart contract will be invoked and among the defined CA depending upon the availability, one CA issue and validate the certificate.

Yakubov et al. [36] proposed a $blockchain-basedPKI$ *managementframework* with the objective to avoid the SPoF limitation of the traditional PKI system. In the developed PKI each CA contains its own smart contract dealing with all

TABLE 3. Related work based on Blockchain clustering.

Ref	Clustering Technique	Key Feature	Blockchain clustering criteria	
			Data clustering	Node clustering
[28]	K-Means	To detect the malicious activity in the bitcoin network.	✓	✗
[29]	K-Means	For node behavioral Analysis.	✓	✗
[30]	BPA	For behavior pattern analysis.	✓	✗
[31]	K-Means	To identify the blockchain data owner.	✓	✗
[32]	K-Means	To detect the fraud transaction.	✓	✗
[33]	AHC	To detect the known and unknown users.	✓	✗

relevant information regarding the certificates including the hash of previously issued or revoked certificates. Qin et al. in [37] proposed a PKI framework *Cecoin* for bitcoin. For issuing the certificate the PoW consensus mechanism is being used. The participating node will try to solve the puzzle or NONCE issued by the initiator. The node solving the puzzle first issues the certificate for the transaction. Tewari et al. [38] proposed *X*.509*Cloud* as the blockchain-based PKI system. The main idea present in this work is to issue different certificates for new requests and the certificate revocation process.

IV. BACKGROUND STUDY

The preliminary study for the current work such as PKI, Blockchain, Clustering, and the need for clustering in blockchain are reported in this section.

A. PKI

The Public Key Infrastructure (PKI) comprises hardware, software, and cryptographic rules to create, store and manage digital certificates for secure internet-based communication [39]. The following are the key elements of the PKI. Figure 1 shows the basic functionality of a PKI system. Before sending data to the receiver, the sender initially requests the *RegistrationAuthority*(*RA*) for issuing a certificate. The *RA* forwards the same request to the *CA* for generating the certificate for the sender. The *CA* sends the certificate along with the private key to the requesting user. After getting the certificate the sender sends the data to the receiver. Simultaneously the *CA* shares the public key of the sender with the *VerificationAuthority*(*VA*). The receiver requests the *VA* for verifying the certificate and after successful verification, the receiver is able to read the data.

- Private and Public Key Pair: PKIs ensure the authenticity, secrecy, and integrity of transactions using asymmetric and symmetric cryptography. Individual end-users, web servers, embedded systems, linked devices, or programs/applications performing business processes might be "Subscribers" in PKI jargon. Asymmetric cryptography gives consumers, devices, or services in an ecosystem a public-private key pair. The group's public key may be used for encryption or digital signature verification. The private key must be kept secret and is only used by its owner for decryption and digital signatures.
- Digital Certificate: It is the credential to verify user identities during a transaction.
- Certificate Authority (CA):The entire stages of certificate management are typically handled by the CA, along with all other facets of certificate administration for a PKI.
- Registration Authority(RA):A registration authority (*RA*) verifies the credentials of the user that requests certificates and then informs the *CA* to issue the same.
- Certificate Revocation List (CRL):A *CRL* is a collection of certificates that have been issued by a CA but had been later revoked by that same *CA* before the certificate expiry. *Delta* and *BaseCRL* are the two different variants of *CRL*. *BaseCRL* is a large list that contains entire revoked certificate details and *DeltaCRL* contains the most recent list of revoked certificates. In each short time interval, the *DeltaCRL* is updated to remove the older revoked certificates.
- Hardware Security Model: It is the optional element for the PKI that helps in safeguarding the key pairs.

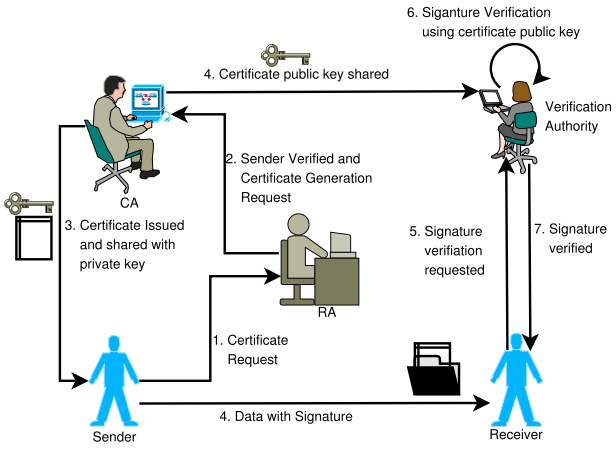


FIGURE 1. Working of a conventional PKI.

B. BLOCKCHAIN

Blockchain technology emerges as the solution to make things decentralized. In the conventional communication system, the entire network needs to depend on a single *CA* for issuing the certificate [40]. The validation of the communication entirely depends upon the trustworthiness of the *CA* which tends to be a single-point failure. To eliminate this

lacuna blockchain technology suggests having multiple *CAs* for multiple transactions which increases the robustness of the network. For the current work, the Ethereum blockchain platform Go Ethereum (*GETH*) [41] is being used.

C. MACHINE LEARNING BASED CLUSTERING AND ITS NEED IN BLOCKCHAIN

Clustering is an unsupervised machine learning technique for making multiple groups based on some similar features [42]. In the case of blockchain, the clustering techniques can be used two different scenarios.

- To make the clusters of blockchain data to identify the malicious activity that occurred in the network.
- To make clusters of the blockchain node to minimize the search space while selecting a *CA* for one transaction.

Clustering the blockchain data is one of the most popular use cases of machine learning-based clustering technique which can be reflected in section V-B and Table 3. The second use case of clustering remains unexplored. A Blockchain network contains multiple number nodes and also multiple transactions. As per the blockchain feature, every transaction must have a different *CA* for validating the transaction. Searching different *CA* every time in the entire blockchain network will take numerous times which can also increase the network overhead. So, clustering the blockchain network emerges as the solution to decrease the network overhead by limiting the search space of *CA* selection. The main issue that node clustering faces are finding the appropriate features for grouping the nodes into different clusters. For this, the response time and validation time can be the two parameters for making the clusters.

V. PROPOSED WORK

This section focuses on the different building blocks of the proposed CTB-PKI system. The proposed CTB-PKI is implemented in the open-source blockchain platform Go Ethereum (*GETH*) with the smart contract as a key element. Figure 2 reflects the block structure of the proposed work. Initially, the node that wants to initiate a transaction has to go for the *CA* selection. For selecting the appropriate *CA* the node needs to select the cluster first. The cluster selection depends on the minimum response time from the transaction initiator node. From the selected cluster a node will be selected as the *CA* depending upon the trust value it has. A node having a higher trust value will have a higher probability to become a *CA*. After selecting the *CA* the certificate is issued to the requesting node and also forwarded the same for the network for synchronizing the same in the *DistributedLedger* (*DLT*).

A. MODEL DESCRIPTION

The proposed CTB-PKI consists of different modules such as the Participant, Validation, and Signature Revocation.

(i) **New Participant:** This module is called when a node wants to communicate in the blockchain. Before communicating the node status in the network is verified.

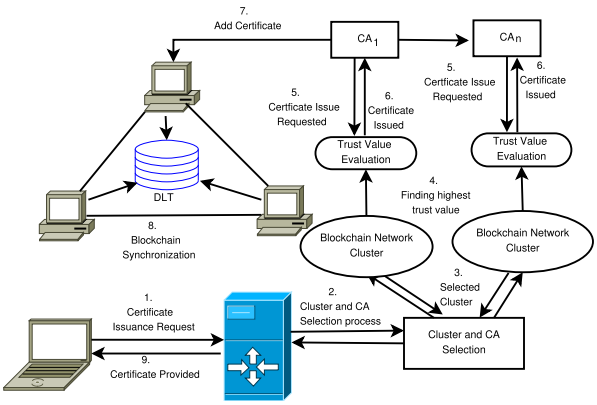


FIGURE 2. Block diagram of the proposed CTB-PKI system.

If the node is found to be a new joinee to the network then the following parameters are invoked. This module has a 7-output tuple for a node when invoked as shown in equation 1.

$$NewParticipant \leftarrow T\{Node_{id}, N_{ETHaddress}, Pr - Key_N, Pu - Key, N_{Expiry}, N_{Rev}\} \quad (1)$$

- $Node_{id}$: It is a random number provided to identify a particular node in the blockchain network.
- $N_{ETHaddress}$: It is an unique *EThereumaddress* provided by the *GETH* environment to a node N.
- $Pr - Key_N$,$Pu - Key_N$: The $Pr - Key_N$ and $Pu - Key_N$ are the private and public keys of a Node (N) to be used during the communication.
- N_{Expiry}: It is the maximum or threshold limit for a node i for which the node N can become a *CA*.
- N_{Rev}: It is a counter of the node N to indicate the number of times a node becomes a *CA*. With the initialization, this counter value is set to 0.

(ii) **Validation:** In this section the validation will be done for two different nodes such as the transaction initiator and selected validator. For instance, node A wants to initiate a transaction with selected CA as B. Then, the input tuple of this module is reflected in equation 2.

$$Validation \leftarrow T\{A_{id}, B_{id}, Exp_A\} \quad (2)$$

A_{id} and B_{id} are verified in or of both transaction initiator and the CA are checked for their existence in the network. Exp_A is another input to this module for verifying the eligibility of a node as *CA*. Both of the conditions are executed in a smart contract. If conditions are satisfied then the node will be allowed to have the corresponding CA for validating the transaction.

(iii) **Signature Revocation:** After every transaction this module will be invoked. Taking the previous instance with A as the initiator and B as the *CA* into consideration the input tuple of this module is shown in equation 3.

$$SignatureRevocation \leftarrow T\{Rev_B, B_{id}\} \quad (3)$$

After every transaction, the revocation id Rev_{id} of the CA will be incremented by 1. Every time the CA's Rev_{id} will be verified against the Exp_i to check the maximum limit of that node for becoming the CA.

B. CLUSTERING

Clustering is a machine learning-based technique that allows making different groups of data points having similar characteristics. The primary objective of the clustering technique is to make an intrinsic grouping of an unlabelled dataset. The main question present behind this technique is how to define the number of clusters. To solve this problem various algorithms including K-Means, K-Means++, DBSCAN, Agglomerative Hierarchical Clustering, etc are present. For the current study, the K-Means (section V-B1) and DBSCAN methods are being used to determine the number clusters.

1) K-MEANS CLUSTERING

K-Means is one of the most popular clustering techniques [43]. The objective of the technique is to find the number of k clusters out of N number of data points. The performance of this algorithm depends upon the optimal k-value selection which is one of the biggest issues of this algorithm. To solve this issue there are several internal validation methods present such as the Elbow Method, Silhouette Coefficient, and Calinski-Harabasz [44]. For the proposed work the Silhouette Coefficient (SC) approach is adopted. To calculate the SC the two attributes response time (RT)and validation time (VT) of the blockchain node are considered. The SC can be calculated by using equation 4. Algorithm 1 shows the pseudocode of the K-Means algorithm.

$$SC = \frac{RT_i - VT_i}{max(RT_i, VT_i)} \tag{4}$$

Algorithm 1 K-Means Clustering Based on$\langle RT_i, VT_i, T_i\rangle$

```
k − max ← 18
k ← 1
while k ≤ k-max do
    CalculateSC
    printSC
    k ← k + 1
end while
Obtain the optimal k with maximum SC value
```

2) DBSCAN

Density-Based Spatial Clustering of application with Noise or DBSCAN technique is used to identify different clusters of the data points that are closed to each other depending on some measurement [45]. It has two inputs mpts and epsilon. Algorithm 2 shows the pseudocode for the DBSCAN method.

- mpts: It is the minimum number of data points required to form a dense region.

TABLE 4. Notations for trust calculation.

Notation	Definition
E_0	Initial Experience level of a new joinee node
E_{min}	The minimum experience level of a node which is set to 0. The experience level is normalized between 0 and 1
E_{max}	Maximum experience level of node
R_{min}	The minimum experience level of a node which is set to 0
α	Feedback score after successful transaction
β	Feedback score after each unsuccessful transaction
E_t	Current experience level of a node n
E_{t+1}	Updated experience level
S_T	Number of Successful transactions
U_T	Number of Unsuccessful Transaction
$S_{T(C1-C2)}$	Successful transaction from cluster C1 to C2
$U_{T(C1-C2)}$	Unsuccessful transaction from cluster C1 to C2

- Epsilon(ϵ): It is the distance measurement that is used to locate the next data points from any random datapoint.

Algorithm 2 DBSCAN Algorithm Based on $\langle RT_i, VT_i, T_i\rangle$

```
Cluster ← φ
for ∀ n ∈ N do
    mark n as visited
    X ← GETNEIGHBOUR(n,ϵ)
    if (|X| <mpts) then
        mark n as the noise
    else
        Cluster← Cluster ∪ n
    end if
end for
```

C. TRUST CALCULATION

Trust is the value that plays a vital role in selecting a particular node as the *CA* for a transaction [46], [47]. The trust (T) of a node can be calculated by two factors including (i) experience level (E) (section V-C1) (ii) reputation factor (R) (section V-C2). Notations used for calculating the trust value are reported in Table 4. The trust value of every participating node is calculated by using the equation 5 and 6 with w_R and w_E as the weight factors such as $w_R + w_E = 1$. Our work considers equal priority on the weightage of experience and reputation parameters.

$$Trust = w_E \times E + w_R \times R \tag{5}$$

$$Trust = \frac{1}{2} \times E + \frac{1}{2} \times R \tag{6}$$

1) EXPERIENCE LEVEL

The experience level (E) is calculated by using positive experience (E_{pos}), and negative experience (E_{neg}). The T_{pos} and T_{neg} are responsible for increasing and decreasing the trust value of a node respectively. The experience level of a node will be updated after every transaction.

1) **Positive Experience:** For a transaction a node n acts as the CA. After the successful transaction the positive experience value (E_{pos}) follows the following linear equation 7.

$$E_{t+1} = E_t + \alpha\Delta \tag{7}$$

where Δ can be defined as the equation 8 with η as the value to normalize the experience value between 0 and 1.

$$\Delta = \eta \times (1 - E_t) \tag{8}$$

2) **Negative Experience:** A node n acts as the CA for a transaction. After every unsuccessful transaction the negative experience value ((E_{neg})) follows the equation 9.

$$E_{t+1} = Maximum(E_{min}, E_t - \beta) \tag{9}$$

2) REPUTATION FACTOR CALCULATION

The reputation factor (R) is the aggregation of intra-cluster trust and inter-cluster trust. In the proposed work network clustering is performed. For a transaction, the CA and the node n can belongs to the same cluster or a different cluster. If both of the nodes belong to a single cluster then the trust is called a direct trust (T_D) otherwise the trust is known as the Indirect trust (T_{ID}). T_D can be calculated as the equation 10:

$$T_D = \begin{cases} Maximum(R_{min}, \dfrac{S_T - U_T}{S_T + U_T}) & \text{if } S_T, U_T \neq 0 \\ 0, & \text{Otherwise} \end{cases} \tag{10}$$

For instance a node i of cluster $C1$ selects a node j as the CA of another cluster $C2$, then the T_{ID} of the node the CA is calculated as equation 11.

$$T_{ID} = \begin{cases} Maximum(R_{min}, \dfrac{S_{T(C1-C2)} - U_{T(C1-C2)}}{S_{T(C1-C2)} + U_{T(C1-C2)}}), \\ \qquad \text{if } S_{T(C1-C2)}, U_{T(C1-C2)} \neq 0 \\ 0, \qquad \text{Otherwise} \end{cases} \tag{11}$$

D. CONSENSUS MODEL

For the proposed $CTB-PKI$ $ProofofAuthority(PoA)$ consensus methodology is adopted. The key concept present behind this consensus method is to choose the CA depending on reputation or trust value. For every successful transaction, the trust value is updated as per section V-C. The reputation of a node as CA will increase for every successful transaction and decrease for every unsuccessful transaction.

E. BLOCKSTRUCTURE

Block is the key element in the blockchain. It is composed of two different components as block header and a body. The block header consists of (i) the hash of the previous block (ii) the time stamp at which the block is created (iii) NONCE which is the optional part that is kept only for the transaction using Proof of Work (PoW) and (iv) the Merkle root which is hash of the root of the Merkle tree. By storing the hash of the previous block the chain of blocks is created which ensures the data integrity. A small change in the transactional data will be reflected as it significantly changes the Merkle root. This also simplifies the transaction verification process by only comparing the generated root hash of the Merkle tree with the stored one. The body of the block indicates the transactional data. Figure 3 shows the block structure used for the current work.

VI. WORKING PRINCIPLE

The core functionality behind the proposed $CTB-PKI$ is to select the CA for a transaction depending on the node trust value. The higher trust value enhances the probability of a node becoming CA. CTB-PKI method suggests an approach for calculating the trust of the nodes (see section V-C). The decentralization characteristic enables the network to have different CA for different transactions. The blockchain network can contain a large number of nodes. So the search space for selecting CA every time increases the computational overhead. To avoid this issue, the proposed work adopts different clustering algorithms such as $K-Means$ and $DBSCAN$ to make different clusters of nodes (see section V-B). Algorithm 3 and Figure 4 show the pseudocode and workflow of the proposed work. The working process of the proposed work is elaborated in the following steps.

Step-1 Initially, the clustering of nodes is executed depending upon two parameters such as $\langle RT, VT \rangle$. It is because the trust value of the participating node is set to 0 initially. The CA selection process can be done based on the input budget $\langle RT, VT \rangle$ by the participant node.

Step-2 After a certain number of transactions, the nodes are re-evaluated for the cluster with an input of 3 values $\langle RT, VT, T \rangle$. Each cluster have average RT and VT value named as RT_{avg} and VT_{avg}. For initiating a transaction the participating node provides a budget of response time RT_{budget} and a budget of validation time VT_{budget}. The cluster which has the least RT and VT compared to RT_{budget} and VT_{budget} is selected as the preferred cluster for our CA selection process. Thereafter all nodes of the selected cluster evaluate their rank by the equation 12.

$$Rank = W_R \times (1 - \frac{RT}{RT_{max}}) + W_V \times (1 - \frac{VT}{VT_{max}}) + W_T \times \frac{T}{T_{max}} \tag{12}$$

Equation 12 has three weighted parameters WR, W_V and W_T which indicate the priority of response time, validation time and trust respectively where $W_R + W_V + W_T = 1$(normalized). The equal priority mode means $W_R = W_V = W_T = \frac{1}{3}$. In general applications, trust and delay are considered fundamental parameters where the delay is $R_T + V_T$. In this sense the equal priority means $W_R + W_V = \frac{1}{2}$ and $W_T = \frac{1}{2}$. The Single priority mode means any one of W_R, W_V, and W_T is unity and the other two are zero; for response time priority, $W_R = 1$, $W_V = 0$, $W_T = 0$; for validation time priority, $W_R = 0$, $W_V = 1$, $W_T = 0$; and for trust priority, $W_R = 0$,

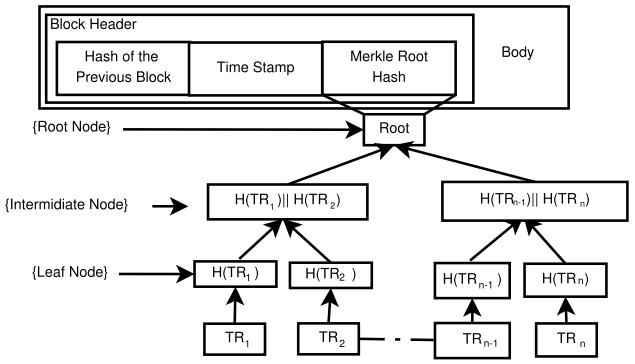


FIGURE 3. Blockstructure of the proposed CTB-PKI system.

$W_V = 0$, $W_T = 1$. The node of the selected cluster which has the maximum rank gets the chance to become the *CA* for the transaction. Algorithm 4 shows the CA selection process. Numerous applications provide arguments for categorizing the weighted priority in the various forms mentioned above. There are many real-time blockchain-based IoT applications like VANET [48] where delay (response time and validation time) plays a very crucial role compared to the trust factor we calculated from previous successful transactions. However, for financial applications trust is a more important issue [49] compared to delay. The three weighted factors can act like a tuning knob, depending on the application these weight values can be changed.

$$Rank = \begin{cases} W_R \times (1 - \dfrac{RT}{RT_{max}}), & \text{if, } W_V, W_T = 0 \\ W_V \times (1 - \dfrac{VT}{VT_{max}}), & \text{if, } W_R, W_T = 0 \\ W_T \times \dfrac{T}{T_{max}}, & \text{if, } W_R, W_V = 0 \\ \dfrac{1}{3} \times RT + \dfrac{1}{3} \times VT + \dfrac{1}{3} \times T & \\ & \text{if,} W_R = W_V = W_T \end{cases} \tag{13}$$

Step-3

The smart contract verifies the selected node N as *CA* by N_{id}, and $N_{ETHaddress}$. If the verification process is successful then the node eligibility for becoming the *CA* is verified.

Step-4 The selected expiry limit N_{Expiry} is compared with the revocation id N_{Rev}. If the N_{Rev} is found smaller than the N_{Rev} then only the *CA* is allowed to validate the transaction. Otherwise, the transaction initiator node is informed to select another *CA*.

Step-5 For every transaction, the CA_{Rev} is incremented by 1. In addition to the CA_{REV}, the trust value of the *CA* is reevaluated (see section V-C). **Step-6** After a certain number of transactions, step 2 is invoked to reform the network cluster.

VII. EMPIRICAL ANALYSIS

This section focuses on the implementation (section VII-A), performance analysis (section VII-B), and time complexity analysis (section VII-C) of the proposed PKI system.

Algorithm 3 Proposed CTB-PKI

```
Initiate Transaction
Invoke Proc K − Means() and DBSCAN()
Define the optimal number of clusters k
Initiate m number of transactions with k clusters
Cluster selection process
Invoke Selection() to select the appropriate CA
Invoke PoA()
for (i=1 to m) do
    get CA_id, CA_ETHaddress, CA_Rev, CA_Expiry
    Invoke Smart Contract to verify the identity of CA
    if (CA_id == N_id) then
        if (CA_ETHaddress== N_ETHaddress) then
            CA Identity verified
        else
            CA identity mismatched. Abort the transaction
            and select a new CA
        end if
    end if
    Invoke Smart Contract to check the eligibility of CA
    if (CA_Rev ≤ N_Expiry) then
        Validate the Transaction
        CA_Rev ++
    else
        Maximum Trial is over for the elected validator.
        Please select another node
    end if
    Calculate Trust of the CA
end for
Invoke K − Means() for reclustering
```

Algorithm 4 CA Selection

```
for (i=1 to k) do
    if (RT_avg(i) < RT_budget & VT_avg(i) < VT_budget) then
        for (j=1 to N) do
            Rank_j = W_R×(1−RT_i/RT_max)+W_V×(1−VT_i/VT_max)+W_T×
            (1 − T_i/T_max)
        end for
    end if
end for
```

A. IMPLEMENTATION

The proposed CTB-PKI is implemented in the open-source Ethereum platform (*GETH*). The *solidity v* 0.4.24 scripting language and *Truffle Suit* are used to deploy the smart contract to the blockchain environment. A system with Windows 10 OS, 8GB RAM, Intel i5 with 2.8 GHz clock speed, 1TB HDD, and 500GB SSD is used to implement the proposed blockchain-based PKI.

B. PERFORMANCE ANALYSIS

The proposed $CTB - PKI$ is implemented in *GETH* with 100 nodes. Each node is associated with 100*ETH* and a 4000000 Gas limit. In Ethereum Ganache truffle suit the

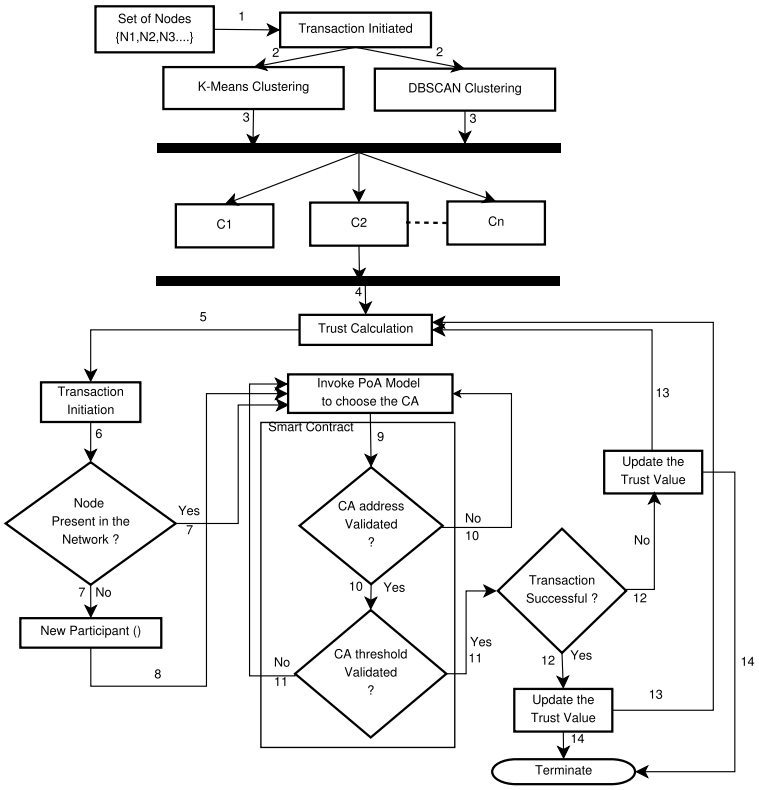


FIGURE 4. Workflow of the proposed CTB-PKI system.

default gas limit for a node is 21000. However, different modules of the proposed PKI framework require more than 21000 gas. Therefore, the gas limit has been changed from the default value to maximum limit. Lowering the gas limit causes a failure in the transaction. Figure 5 and 6 shows the cluster formulation using the $K-Means$ and $DBSCAN$ algorithm with $\langle RT, VT \rangle$ as the input parameter respectively. 17 iterations starting from 2 to 18 are performed for both the clustering algorithm to calculate the SC. The cluster that has the highest SC is considered the optimal number of clusters. From experimental work, it is observed that the SC value for the number is $\sim$ 0.56 and $\sim$ 0.43 for $K-Means$ and $DBSCAN$ respectively. So for the current work, the optimal number of clusters is taken as 2. Figure 7 shows the number of clusters with RT, VT, and T as the input variable. Due to high computational time the $DBSCAN$ algorithm is not used further. SC value for $k = 2$ is ~ 0.61 which is highest in contrast to other k value. The number of elements in clusters 0 and 1 is 61 and 39 respectively. Table 5 shows the number of nodes present in each cluster with different clustering algorithms.

Figure 8 and Figure 9 reflect the response time and validation time of the proposed PKI with and without the clustering algorithm respectively. From the figure 9, it can be observed that the proposed work reduces RT about $\sim$ 38.2%. This improvement is due to the reduction of search space in the CA selection process. Figure 8 shows an improvement of VT with clustering in contrast to the VT without clustering. The

TABLE 5. Number of nodes in each cluster.

Cluster	Algorithm		% of total node	
	K-Means	DBSCAN	K-Means	DBSCAN
Cluster 0	72	65	72 %	65%
Cluster 1	28	35	28%	35%
Total nodes	100	100	100%	100%

proposed work reduces the VT about $\sim$ 2.2%. This improvement is because of the implication of trust value in selecting the CA for validating the transaction. Figure 10 shows the gas utilization with the different number of transactions of the proposed work. The average gas utilization of the proposed work is approximately 5×10^4.

The participant node needs to set its own budget by setting the corresponding weight factors W_R, W_V, and W_T. Depending upon the input the CA is selected with appropriate RT, VT, and T. Table 6 shows the CA selection ranking process for 10 transaction with different input budget. If the W_R is set to 1 then the node having the lowest RT value is considered as the CA. Accordingly, if the W_T is set to 1, then the node having the highest trust value within the cluster is selected as the CA.

C. TIME COMPLEXITY ANALYSIS

The proposed $CTB-PKI$ has different executable modules such as *New Participant*, *Validation*, *Signature Revocation*,

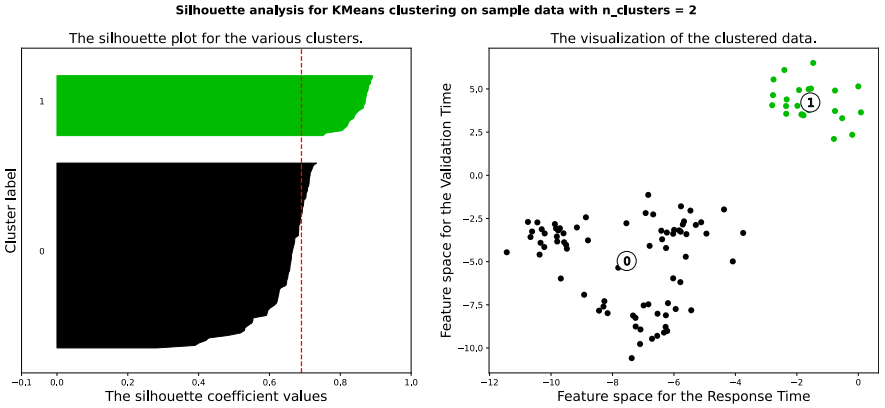


FIGURE 5. Number of Cluster using K-Means.

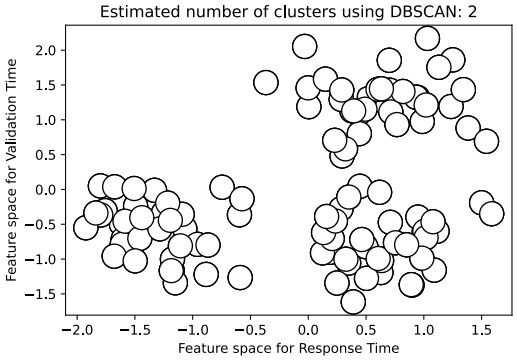


FIGURE 6. Number of clusters using DBSCAN algorithm.

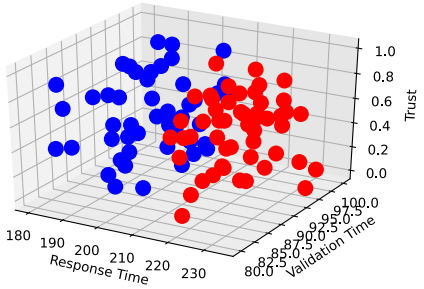


FIGURE 7. K-Means clustering with RT, VT, and Trust as the feature.

TABLE 6. CA selection ranking based on the selected input budget.

Transaction	Input Budget	RT (in Sec)	VT (in Sec)	T
T_1	$\langle 1,1,0\rangle$	106	67	0.7
T_2	$\langle 1,0,0\rangle$	111	71	0.47
T_3	$\langle 0,0,1\rangle$	146	81	0.95
T_4	$\langle 1,0,1\rangle$	112	79	0.85
T_5	$\langle 0,0,1\rangle$	142	74	0.94
T_6	$\langle 1,1,0\rangle$	108	69	0.37
T_7	$\langle 0,0,1\rangle$	137	71	1
T_8	$\langle 1,1,1\rangle$	112	69	1
T_9	$\langle 1,0,0\rangle$	114	70	0.52
T_{10}	$\langle 1,1,1\rangle$	119	75	0.81

Smart Contract,$K-Means$, and *DBSCAN*. Among these the *SmartContract* and *NewParticipant* modules have the

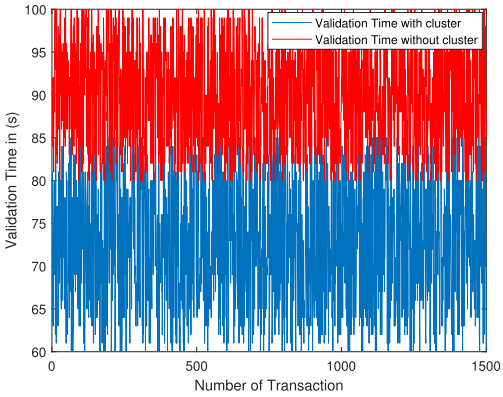


FIGURE 8. Validation time with and without cluster.

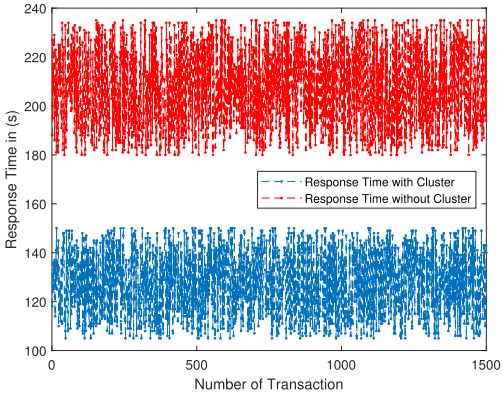


FIGURE 9. Response time with and without cluster.

time complexity of $O(n)$. Whereas the other two modules have constant time complexity $O(1)$. *NewParticipant* and *SmartContract* may receive multiple transactions thus making the time complexity of these two modules as $0(n)$. While for the other two modules *Validation* and the *SignatureRevocation* no transactional messages are generated, thus making the complexity of these two modules as $O(1)$. Implementing the PoA consensus mechanism has the time complexity $O(logn)$. Finally, the time complexity of

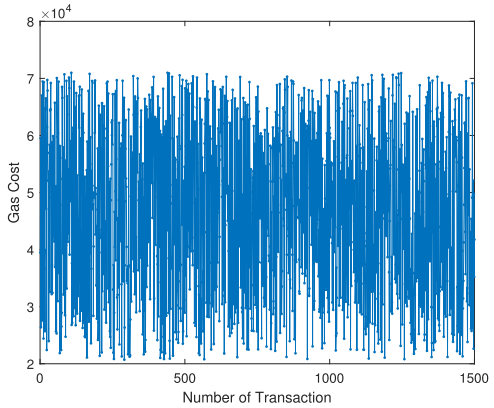


FIGURE 10. Gas utilization for different transactions.

$K-Means$ and $DBSCAN$ algorithm are $O(kN)$ and $O(N^2)$ with N as the number of nodes present in the network. The time complexity of each individual module is reported in Table 7.

TABLE 7. Time Complexity Analysis of proposed CTB-PKI model.

Module	Time Complexity
New Participant	$O(n)$
Validation	$O(1)$
Signature Revocation	$O(1)$
Smart Contract	$O(n)$
PoA	$O(logn)$
K-Means	$O(kN)$
DBSCAN	$O(N^2)$

D. CRITICAL ANALYSIS

$BB-PKI$ in [34] proposed a PKI with the objective to avoid the $SPoF$ issue of the conventional PKI systems by introducing RAs. The node that wants a certificate for communication needs to forward the request to RA. RA then forwards the same request to the corresponding CA. With this solution, the proposed methods put a limitation on the P2P network concept. In $BlockPKI$ [35] a group of nodes is defined for becoming CA. For every transaction, the node that belongs to that group only can have the chance for becoming the CA which makes the whole process semi-decentralized.

In $Blockchain-basedPKImanagementframework$ [36] CA needs to store all the relevant information regarding the certificate issuance and revocation. This process needs high memory availability at the CA end which becomes the main issue of the proposed system. In $Cecoin$ [37] the main issue is the adoption of PoW consensus mechanism for selecting the CA. PoW needs high computational capability at the node end which becomes the main issue for the lightweight clients in participating in the network communication. The limitation present behind the $X.509Cloud$ [38] is the number of certificates during each transaction as this PKI generates the different certificates for transaction and revocation. Table 8 shows the overall comparison of the above-mentioned blockchain-based PKIs in contrast to the proposed PKI system.

TABLE 8. Comparison of the proposed work with existing literature.

PKI	Registration	Validation	Revocation	Trust Calculation	Node Clustering
[34]	✓	✓	✓	✗	✗
[35]	✓	✗	✓	✗	✗
[36]	✓	✓	✓	✗	✗
[37]	✓	✓	✓	✗	✗
[38]	✓	✓	✓	✗	✗
Proposed Work	✓	✓	✓	✓	✓

VIII. CONCLUSION

The proposed work addresses the limitation of the computational overhead of the existing PKI systems. This work reports a blockchain-based PKI system $CTB-PKI$ which uses clustering algorithms K-Means and DBSCAN to reduce the CA search space. The time complexity analysis shows that the K-Means algorithm is more suitable compared to the DBSCAN method for the current work. This work also focuses on the trust calculation of every participating node. The node, having a higher trust value and lower validation time, and lower response time has a higher probability of becoming the CA for a transaction. For every successful transaction, the CA trustworthiness is increased and the trust value is decreased for every unsuccessful transaction. The performance of the proposed system is evaluated based on the response time, validation time, and gas utilization required for different transactions. The result analysis shows that network clustering puts an impact on response time and validation time. The proposed approach reduces ∼38.5% response time and ∼2.2% validation time compared to the PKI systems without clustering. The improvement in response time and validation time reduces transaction validation turnaround time in a blockchain-based communication system which makes the proposed $CTB-PKI$ more suitable for Blockchain 2.0 and 3.0 applications.

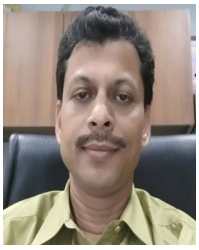
In our proposed $CTB-PKI$, the trust of every node is calculated based on successful and unsuccessful transactions. Other node communication quality factors such as data transmission rate and data delay rate can be considered for making the trust calculation more effective. The inclusion of clustering may increase the network performance by decreasing the latency such as RT and VT. However, the network energy consumption and computation effort have not been studied meticulously in our paper which we intend to address in our future studies. Moreover, we intend to improve the trust value of the proposed $CTB-PKI$ as well.

REFERENCES

[1] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2027–2051, 3rd Quart., 2016.
[2] M. E. Hellman, "An overview of public key cryptography," *IEEE Commun. Mag.*, vol. 40, no. 5, pp. 42–49, May 2002.
[3] A. Alrawais, A. Alhothaily, X. Cheng, C. Hu, and J. Yu, "SecureGuard: A certificate validation system in public key infrastructure," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5399–5408, Jun. 2018.

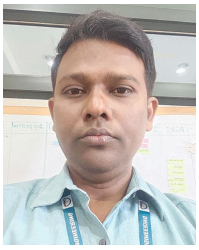
[4] A. P. Felt, R. W. Reeder, A. Ainslie, H. Harris, M. Walker, C. Thompson, M. E. Acer, E. Morant, and S. Consolvo, "Rethinking connection security indicators," in *Proc. 12th Symp. Usable Privacy Secur. (SOUPS)*, 2016, pp. 1–14.
[5] E. Schechter, "Moving towards a more secure web," Google Security Blog, Tech. Rep., 2016.
[6] T. Vyas, "No more passwords over http, please," Mozilla Blog, Tech. Rep., 2016.
[7] R. Holz, J. Amann, O. Mehani, M. Wachs, and M. A. Kaafar, "TLS in the wild: An internet-wide analysis of TLS-based protocols for electronic communication," 2015, *arXiv:1511.00341*.
[8] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X. 509 internet public key infrastructure online certificate status protocol-OCSP," Tech. Rep., 2013.
[9] S. Garfinkel, "Pretty good privacy (PGP)," in *Encyclopedia of Computer Science*. U.K.: Wiley, 2003, pp. 1421–1422.
[10] B. Laurie, A. Langley, and E. Kasper, "Certificate transparency," Tech. Rep., 2013.
[11] L. Chuat, P. Szalachowski, A. Perrig, B. Laurie, and E. Messeri, "Efficient gossip protocols for verifying the consistency of certificate logs," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Sep. 2015, pp. 415–423.
[12] T. H.-J. Kim, L.-S. Huang, A. Perring, C. Jackson, and V. Gligor, "Accountable key infrastructure (AKI): A proposal for a public-key validation infrastructure," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 679–690.
[13] D. Basin, C. Cremers, T. H.-J. Kim, A. Perrig, R. Sasse, and P. Szalachowski, "ARPKI: Attack resilient public-key infrastructure," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 382–393.
[14] J.-G. Dumas, P. Lafourcade, F. Melemedjian, J.-B. Orfila, and P. Thoniel, "LocaLPKI: An interoperable and IoT friendly PKI," in *Proc. Int. Conf. E-Business Telecommun.* Cham, Switzerland: Springer, 2017, pp. 224–252.
[15] M. A. Vigil, C. T. Moecke, R. F. Custodio, and M. Volkamer, "The notary based PKI," in *Proc. Eur. Public Key Infrastruct. Workshop*. Cham, Switzerland: Springer, 2012, pp. 85–97.
[16] Y. Wang and J. Vassileva, "Bayesian network trust model in peer-to-peer networks," in *Proc. Int. Workshop Agents Comput.* Cham, Switzerland: Springer, 2003, pp. 23–34.
[17] B. Yu and M. P. Singh, "An evidential model of distributed reputation management," in *Proc. 1st Int. Joint Conf. Auto. Agents Multiagent Syst.*, 2002, pp. 294–301.
[18] A. Yamamoto, D. Asahara, T. Itao, S. Tanaka, and T. Suda, "Distributed pagerank: A distributed reputation model for open peer-to-peer network," in *Proc. Int. Symp. Appl. Internet Workshops. Workshops.*, 2004, pp. 389–394.
[19] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, Jul. 2004.
[20] H. Wen, X. Ren, and G. Xu, "A DS evidence theory based trust model for the P2P network," *J. Xi'An Univ.*, vol. 32, no. 3, pp. 400–402, 2005.
[21] S. Song, K. Hwang, R. Zhou, and Y.-K. Kwok, "Trusted P2P transactions with fuzzy reputation aggregation," *IEEE Internet Comput.*, vol. 9, no. 6, pp. 24–34, Nov. 2005.
[22] R. Zhou and K. Hwang, "PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 4, pp. 460–473, Apr. 2007.
[23] Y. Sun, Q. Zhao, and P. Zhang, "Trust degree calculation method based on trust blockchain node," in *Proc. IEEE Int. Conf. Service Oper. Logistics, Informat. (SOLI)*, Nov. 2019, pp. 122–127.
[24] J. Ahn, M. Park, H. Shin, and J. Paek, "A model for deriving trust and reputation on blockchain-based E-payment system," *Appl. Sci.*, vol. 9, no. 24, p. 5362, Dec. 2019.
[25] W. She, Q. Liu, Z. Tian, J.-S. Chen, B. Wang, and W. Liu, "Blockchain trust model for malicious node detection in wireless sensor networks," *IEEE Access*, vol. 7, pp. 38947–38956, 2019.
[26] B. Zhao, Y. Liu, X. Li, J. Li, and J. Zou, "TrustBlock: An adaptive trust evaluation of SDN network nodes based on double-layer blockchain," *PLoS ONE*, vol. 15, no. 3, Mar. 2020, Art. no. e0228844.
[27] Y. Inedjaren, M. Maachaoui, B. Zeddini, and J.-P. Barbot, "Blockchain-based distributed management system for trust in VANET," *Veh. Commun.*, vol. 30, Aug. 2021, Art. no. 100350.
[28] F. Zola, M. Eguimendia, J. L. Bruse, and R. O. Urrutia, "Cascading machine learning to attack bitcoin anonymity," in *Proc. IEEE Int. Conf. Blockchain*, Jul. 2019, pp. 10–17.
[29] S. S. Chawathe, "Clustering blockchain data," in *Clustering Methods for Big Data Analytics*. Berlin, Germany: Springer, 2019, pp. 43–72.
[30] B. Huang, Z. Liu, J. Chen, A. Liu, Q. Liu, and Q. He, "Behavior pattern clustering in blockchain networks," *Multimedia Tools Appl.*, vol. 76, no. 19, pp. 20099–20110, Oct. 2017.
[31] D. Ermilov, M. Panov, and Y. Yanovich, "Automatic bitcoin address clustering," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 461–466.
[32] M. Harrigan and C. Fretter, "The unreasonable effectiveness of address clustering," in *Proc. Int. IEEE Conf. Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People, Smart World Congr. (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, Jul. 2016, pp. 368–373.
[33] M. Fleder, M. S. Kester, and S. Pillai, "Bitcoin transaction graph analysis," 2015, *arXiv:1502.01657*.
[34] A. Garba, Q. Hu, Z. Chen, and M. R. Asghar, "BB-PKI: Blockchain-based public key infrastructure certificate management," in *Proc. IEEE 22nd Int. Conf. High Perform. Comput. Commun., IEEE 18th Int. Conf. Smart City, IEEE 6th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2020, pp. 824–829.
[35] L. Dykcik, L. Chuat, P. Szalachowski, and A. Perrig, "BlockPKI: An automated, resilient, and transparent public-key infrastructure," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2018, pp. 105–114.
[36] A. Yakubov, W. M. Shbair, A. Wallbom, D. Sanda, and R. State, "A blockchain-based PKI management framework," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp.*, Tapei, Tawain, Apr. 2018, pp. 1–6.
[37] B. Qin, J. Huang, Q. Wang, X. Luo, B. Liang, and W. Shi, "Cecoin: A decentralized PKI mitigating MitM attacks," *Future Gener. Comput. Syst.*, vol. 107, pp. 805–815, Jun. 2020.
[38] H. Tewari, A. Hughes, S. Weber, and T. Barry, "X509Cloud—Framework for a ubiquitous PKI," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2017, pp. 225–230.
[39] J. Weise, "Public key infrastructure overview," *Sun BluePrints OnLine*, pp. 1–27, Aug. 2001.
[40] S. S. Gupta. (2017). *Blockchain*. IBM. [Online]. Available: http://www.IBM.COM
[41] *Go Ethereum*, 2020.
[42] T. S. Madhulatha, "An overview on clustering methods," 2012, *arXiv:1205.1117*.
[43] K. P. Sinaga and M.-S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020.
[44] J. Baarsch and M. E. Celebi, "Investigation of internal validity measures for K-means clustering," in *Proc. Int. Multiconf. Eng. Comput. Scientists*, vol. 1, 2012, pp. 14–16.
[45] S. U. Rehman, S. Asghar, S. Fong, and S. Sarasvady, "DBSCAN: Past, present and future," in *Proc. 5th Int. Conf. Appl. Digit. Inf. Web Technol. (ICADIWT)*, Feb. 2014, pp. 232–238.
[46] N. Truong, G. M. Lee, K. Sun, F. Guitton, and Y. Guo, "A blockchain-based trust system for decentralised applications: When trustless needs trust," *Future Gener. Comput. Syst.*, vol. 124, pp. 68–79, Nov. 2021.
[47] W. Hao, J. Zeng, X. Dai, J. Xiao, Q.-S. Hua, H. Chen, K.-C. Li, and H. Jin, "Towards a trust-enhanced blockchain P2P topology for enabling fast and reliable broadcast," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 2, pp. 904–917, Jun. 2020.
[48] Y. Zhang, F. Tong, Y. Xu, J. Tao, and G. Cheng, "A privacy-preserving authentication scheme for VANETs based on consortium blockchain," in *Proc. IEEE 92nd Veh. Technol. Conf. (VTC-Fall)*, Nov. 2020, pp. 1–6.
[49] D. Boughaci and A. A. Alkhawaldeh, "Enhancing the security of financial transactions in blockchain by using machine learning techniques: Towards a sophisticated security tool for banking and finance," in *Proc. 1st Int. Conf. Smart Syst. Emerg. Technol. (SMARTTECH)*, 2020, pp. 110–115.

AMRUTANSHU PANIGRAHI (Member, IEEE) received the B.Tech. degree from BPUT, Odisha, and the M.Tech. degree in information technology from the College of Engineering and Technology, Government of Odisha. He is currently pursuing the Ph.D. degree with the Department of CSE, SOA University, Bhubaneswar. He is currently working as a Research Scholar with the Department of CSE, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India.



AJIT KUMAR NAYAK (Member, IEEE) received the degree in electrical engineering from the Institution of Engineers, India, in 1994, and the M.Tech. and Ph.D. degrees in computer science from Utkal University, in 2001 and 2010, respectively. He is currently the Professor and the HoD of the Department of Computer Science and Information Technology, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha. He has coauthored a book *Computer Network Simulation using NS2* (CRC Press). He has published about 55 research papers in various journals and conferences. His research interests include computer networking, ad hoc and sensor networks, machine learning, natural language computing, speech, and image processing. He has also participated as an organizing member of several conferences and workshops at the international and national levels.

ROURAB PAUL (Member, IEEE) received the B.Sc. and M.Sc. degrees in electronic science and the Ph.D. degree in cryptography and related VLSI design from the University of Calcutta, in 2008, 2010, and 2017, respectively. He is currently an Assistant Professor with Siksha 'O' Anusandhan University, Bhubaneswar, Odisha. Previously, he was a Postdoctoral Fellow at the Computer Science and Engineering Department, Indian Institute of Technology Kanpur. He was a Senior Research Fellow at the School of I.T., Calcutta University. He also worked for the European Organization for Nuclear Research (CERN), Geneva, Switzerland, in the Large Ion Collider Experiment (ALICE), from 2015 to 2016. He held visiting research fellow position in the Electronics and Communication Engineering Department, National University of Singapore (NUS), from September 2013 to January 2014. He has been engaged in teaching, research, and industrial consultancy, from 2010 to 2012. He was a Visiting Lecturer at the Acharya Prafulla Chandra College, Kolkata, and Techno India, Kolkata. He was a Senior Academic Consultant in Convergent Solutions, and also joined an Internship Program in i-cee Design Technology, Kolkata.



BIBHUPRASAD SAHU (Member, IEEE) received the B.Tech. degree in information technology from SSIET, Chennai, and the M.Tech. degree in CSE from the National Institute of Science and Technology. He is currently an Assistant Professor with the AI and DS Graduate Program, Vardhaman College of Engineering, Hyderabad. His research interests include the application of evolutionary algorithms for disease diagnosis and machine learning.

SHASHI KANT (Member, IEEE) received the Master of Business Administration degree from UGC-NET, the Master of Sociology degree from UGC-JRF, ANC-NIIT, and the Ph.D. degree in management from India. He is currently working with the Department of Management, Bule Hora University, Ethiopia, Horn of Africa. He has over 15 years of teaching and research experience in the field of management and marketing in India and Ethiopia. He has published several research papers in Scopus, WOS, and PubMed reviewed international journals. He already published books on *Strategic Management* and *Entrepreneurship and Perspectives of Marketing*. Recently his book *Computer Applications in Engineering and Management* (CRC Press), Taylor, and Francis. He has taught several courses in management like information systems, system analysis and design, strategic management, and entrepreneurship development from fundamental to advanced levels at higher educational institutions. His commitment and approach to teaching have been rewarded with the highest teaching evaluations an instructor can receive: in an institution particularly dedicated to outstanding teaching, he is consistently among the top 5% of all teaching staff. He has held numerous previous administrative positions, including reviewer, and entrepreneurship trainer (Ministry of Skill Development, India). He has also been involved in developing and implementing UG and PG curriculums of strategic management and entrepreneurship development. His research interests include management, information systems, strategic management, marketing, and entrepreneurship development.