IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 11 April 2022, accepted 4 June 2022, date of publication 21 June 2022, date of current version 27 June 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3184799

Introduction of Formal Methods in Blockchain Consensus Mechanism and Its Associated Protocols

SUDHANI VERMA, DIVAKAR YADAV, AND GIRISH CHANDRA
Department of Computer Science and Engineering, Institute of Engineering and Technology, Lucknow 226021, India
Corresponding author: Sudhani Verma (2307@ietlucknow.ac.in)

ABSTRACT As the size of data is increasing exponentially, its security is a major concern. Emerging technology like blockchain is used to provide security to systems. Since the inception of blockchain, it has been adopted by researchers and industry both, however, it gained enormous attention after cryptocurrency. It can be defined as a means of storing information in such a way that modification and hacking the system is difficult or impossible. A blockchain is a decentralized ledger that is digital and public, consisting of records of transactions called blocks. A consensus technology assures that all nodes agree on a unique sequence for appending blocks. A comprehensive examination of these algorithms will aid in understanding how and why each blockchain operates in the manner that it does. In this study, we addressed extensively used consensus techniques in the blockchain and the importance of consensus protocol in blockchain technology. The underlying consensus algorithm is a critical component of every blockchain-based system which determine the performance and security of the system. Ensuring the correctness of consensus protocols is uttermost important to create trust in the blockchain-based systems and formal methods are the way to create that trust and develop correct and verified systems. Formal modeling is a method of writing a system mathematically and examining the correctness and verifying the developed system. This study analyzed the importance of consensus mechanisms and how formal methods are helping to develop a correct blockchain-based system. The current scenario of the application of formal methods in the consensus mechanism of blockchain for their verification is presented.

INDEX TERMS Blockchain, consensus protocols, distributed ledger technology, formal methods, formal verification.

I. INTRODUCTION

The breakthrough technology blockchain was initially introduced by Stuart Haber & W. Scott Stornetta in 1991 [1], but after the introduction of Bitcoin in 2009 [2] it gains enormous attention. A blockchain is a digital ledger that is decentralized and constantly updated by several nodes utilizing distributed cryptography technology without the use of a centralized authority. It is also referred to as distributed ledger technology (DLT) due to its decentralized nature; both are used as synonyms in literature. Though blockchain technology gains attention due to cryptocurrency it is no limit to it. The usage of blockchain in the different sectors is also increasing day by day [3]. The growth of data from various sensors, social data, transaction data, etc. is exponenetial. As per a report from International Data Corporation (IDC) [4], 64.2ZB of data was created in 2020 and the security of this exponentially generated data has become one of the critical concerns. To provide security to various safety-critical and time-sensitive systems, the conventional central server concept is incapable. Therefore, a decentralized system where non-trusting members can interact with each other without a centralized trusted third party is the need of the hour. The emerging technology blockchain has the potential to provide all the needed requirements. The information that will be included in blockchain is authenticated by all the participated nodes. All nodes must agree on a unique sequence, in which the items will be added in the chain with the help of a consensus process. The blockchain consensus protocol is primarily used to verify that the records are accurate and trustworthy. It is also used to

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

create coherence among the nodes participated in appending the new block into the existing blockchain. Blockchain has addressed the challenge of transitioning from a low-trust centralized single third-party ledger to a high-trust decentralized ledger owned by several organizations, or in other words, verifying nodes [5]. The consensus method, which determines how the agreement is established among all the nodes to add a new block, is an essential contribution to blockchain's work. The consensus mechanism can be referred to as the backbone of the blockchain-based system in generating trust and providing fault tolerance in the system. We discuss a wide spectrum of consensus protocols in blockchain technology by presenting the types of protocols, their working mechanism, and potential drawbacks of each protocol such as the processing power and convergence time. We also describe the prime advantages of the underlying consensus algorithms for example the integrated security protocols and tolerance to various threats, these benefits increase its prospective usage in other sectors. Consensus techniques for tolerating byzantine errors have resurfaced as a result of their use in blockchain systems. As consensus protocols play a pivotal role in the blockchain-based system, their correct functioning is paramount important. An introduction of formal methods in this type of emerging technology can ensure the correct working of the underdeveloped system. Also, formal methods serve as the tool for developing and writing correct system specifications for a long time, the application of formal methods at the beginning of any technology is more beneficial for improving the effects of emerging technology in the IT world. As blockchain technology is in its nascent state, not much research has been conducted on the application of formal methods in blockchain and its underlying protocols. The application of formal methods in blockchain technology is to verify the overall behavior of the blockchain-based system and it can be applied to the various underlying protocol such as cryptographic, consensus, and security protocols. This study mainly focuses on the application of formal methods in consensus protocols and aims to provide answers to the following questions:

- What is the role of different consensus mechanisms in building trust in a blockchain-based system?
- Why formal methods are applied in consensus protocols of blockchain-based systems like smart contracts and their current scenario.

The main contribution of this article is that it discusses the importance and present scenario of formal methods in consensus mechanism used in blockchain-based systems. However, we have not found many surveys related to this topic. We attempt to provide a clear picture of formal languages used for writing the specifications and the verification techniques used to verify the blockchain-based system or smart contracts. This article is partitioned into six sections, section 1 is an introduction which includes the objective of this study, and a brief introduction about the topics discussed in the article. Section 2 consists of an overview and working of blockchain technology. In section 3, we discuss different consensus mechanisms with their usages in different types of blockchain and their limitations. Section 4 outlines the introduction of formal methods in consensus algorithms. The next section discuss the various techniques which can be applied in the blockchain, also includes some literature reviews related to formal modeling of consensus protocols. In the last section, we summarize the article by providing future aspects of formal modeling of consensus protocols in the blockchain-based system.

II. FUNDAMENTALS OF BLOCKCHAIN AND SMART CONTRACTS

Blockchain is a distributed ledger of immutable transactions shared across all the participating nodes. The blockchain uses cryptographic hashes for verification of authenticity data. The blocks are added using atomic broadcast to maintain the order. At each node, a copy of the blockchain is maintained and to keep the global ledger consistent a consensus mechanism is used whenever a new transaction is executed and a new block is created. The cryptographic hashes are used as a fingerprint that verifies the data, any modification in data create a new hash which mismatches with the original fingerprint. All the transactions with their hashes are grouped in blocks. Each block will have a combined hash of all these transactions, this hash act as a fingerprint of the block. These hashes are used to create links between the blocks by storing the hash of the previous block in the header of the new subsequent block. As a result of this, a chain of cryptographically secured blocks is created that consists of all the information, referred to as a blockchain. Fig. 1 represents the block structure in a bitcoin blockchain, the block is consisting of a header and a transactions list of that block. Inside the header, the block is having the following six fields. (i) Timestamp when the block is created. (ii) Previous blocks' hash is saved to provide a link between blocks in the blockchain. (iii) Merkle Root also referred to as transaction root consists of the hash value of all validated transactions. The hash value of each valid transaction is calculated and thereafter this hash value of each transaction is pairwise combined with other transactions' hash values, and another hash is generated. This process is repeated until we get one hash generated from all the transactions. The process is implemented using the Merkle tree. (iv) Version of the protocol used by the node which is proposing the new block. (v) Nonce is a result of solving a mathematical puzzle that is computationally very hard, used in the PoW consensus protocol. (vi) Bits are used to indicate the difficulty level of the PoW. The idea of blockchain was first coined by Harber and Scott in 1991 [1], introducing the concept to find a solution for the security of documents so that documents cannot be modified or tampered with. It was proposed that a time-stamped document can be stored in a cryptographically secured chain. Later, in 1992 with the introduction of Merkle trees, the system becomes more efficient by storing more documents in one block. However, the technology went unused until its first implementation in 2008, Bitcoin [2]. It gave an enormous

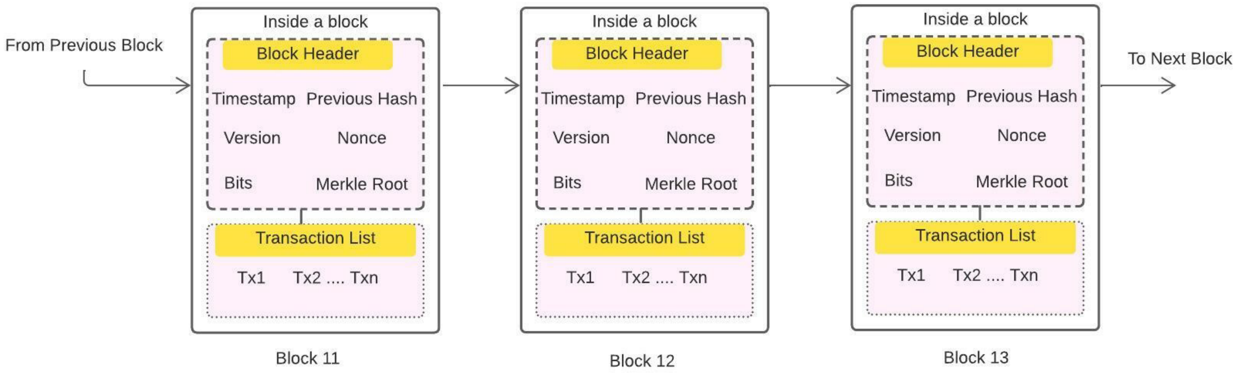

FIGURE 1. Block Structure of Bitcoin Blockchain.

jump in use of blockchain technology. Along with blockchain technology, the second most widely used technology in bitcoin is the Hashcash, Proof of Work (PoW) algorithm, known as PoW mechanism. This decentralized P2P protocol is used to provide protection against double-spending problems by tracking and verifying the transactions. Later, in 2013, Vitalik Burterin started working on a scripting language for bitcoin. He developed a new blockchain-based distributed computing platform with a scripting facility called Ethereum [6]. These are also called Smart Contracts [7]; it is defined as the programs and scripts that are deployed on the blockchain. A smart contract initiates a multi-step process automatically as soon as the predefined conditions are satisfied. Earlier the concept of a smart contract is a script stored on the blockchain is introduced by Nick Szabo in 1994 [8]. Smart Contract resides on chain and has a unique address, they can be initiated by addressing a transaction to it. Once the details of the triggering transaction are mentioned, the smart contract will execute automatically and independently on every node in the network. Therefore, we can say that every node in a smart contract runs a virtual machine (VM). In Ethereum blockchain, a virtual machine referred to as EVM is provided. Smart contract transactions are traceable, transparent, and irreversible. The evolution of blockchain can be summarized below in Fig.2.

In today's scenario, blockchain technology is gaining a lot of mainstream attention and its usage is not limited to cryptocurrency [9]. The formalization and verification of blockchain's safety and security features are becoming increasingly important as it finds applications in a wide range of domains with high-assurance needs. The blockchain network is a good example of a high-confidence distributed communication technology. As a result, formal approaches can help to enhance the security and overall credibility of the blockchain-based system for users.

A. WORKING OF BLOCKCHAIN

The basic important technical components for the proper working of blockchain are Hash Chain, Merkle tree and Digital Signature [10]. The blockchain is primarily organized using three components. (i) Hash pointers, which are used to indicate the location of data. These hash pointers are also used to identify any modification in the data. If user wants to modify the stored data, it has to update the hash pointer of all the previous blocks. (ii) Merkle Tree, it is a binary search tree, all the hash pointers are linked with the help of the Merkle tree. Merkle tree has the potential to keep the data secure and restrict any attempt of modification. (iii) Digital signature, it is a cryptographic algorithm used to establish the validity of data. It is also a technique for ensuring that data hasn't been tampered with. A digital signature must be verifiable and unforgeable. In the context of decentralized blockchain, the consensus is required for appending a new block into the existing global chain. Whenever a new block is created, it is broadcasted to the network, each node can choose whether or not to include the newly created block in its copy of the global ledger. Consensus is used to get an agreement by majority of nodes in the network on a single state modification in order to protect the extension of the global ledger i.e., blockchain. The combination of hash chain, digital signature, Merkle tree, and consensus protocols together help to create the trust in blockchain enabled system. If a node wants to initiate a transaction, it will create the transaction, the transaction will wait in a transaction pool until it has not been added to the blockchain. Prior to adding to the blockchain it must be approved by majority of nodes. The other nodes collect the transaction from the pool, validate the transaction and add it to a block, this block will be further broadcasted in the network, as depicted in Fig. 3. The other nodes verify the block with the help of a consensus mechanism [11]. A consensus mechanism plays a pivotal role in the creation of blockchain as it is responsible for validation and verification of the transactions. A consensus protocols make important decisions in appending the blocks in the blockchain.

B. TYPES OF BLOCKCHAIN

Every blockchain is comprised of nodes connected via a peer-to-peer (P2P) network. The shared ledger is copied

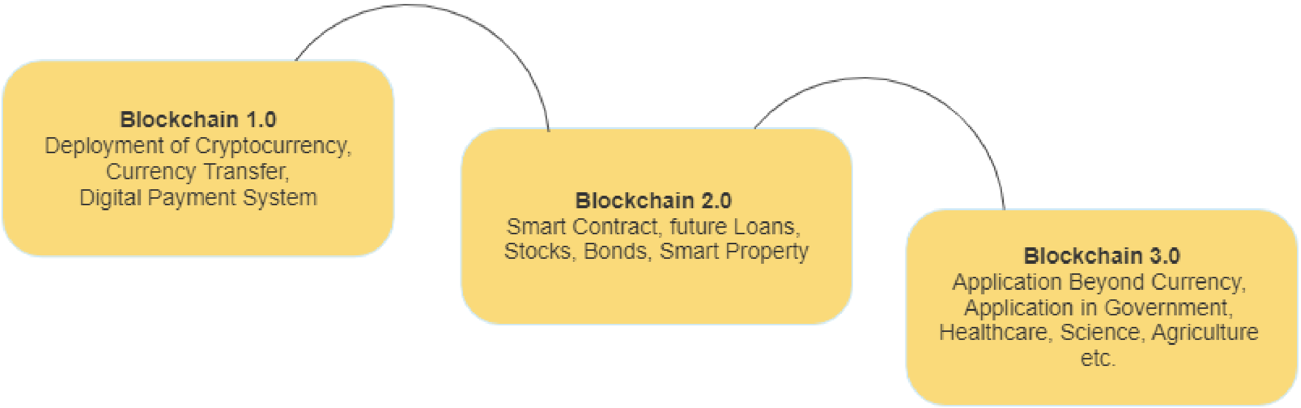
Blockchain 1.0
Deployment of Cryptocurrency,
Currency Transfer,
Digital Payment System
Blockchain 2.0
Smart Contract, future Loans,
Stocks, Bonds, Smart Property
Blockchain 3.0
Application Beyond Currency,
Application in Government,
Healthcare, Science, Agriculture
etc.

FIGURE 2. Evolution of Blockchain Technology.

in every network node, which is regularly updated. Nodes may verify transactions, transmit and receive messages, and generate blocks also capable of validating transactions, sending and receiving messages, and generating blocks. Blockchain has been divided majorly into the following types based on the process of participation of nodes in consensus blockchain [12].

1) PUBLIC BLOCKCHAIN

There is no centralized authority which have more power in the network than others. Anyone can join or leave the network as per their choice. The blockchain is available to the public, and any participating node can validate a transaction. Bitcoin is an example of a public blockchain. In the case of bitcoins, the transaction is validated by miners. They receive bitcoins in the form of transaction fees as well as fresh bitcoins generated as a result of their efforts in solving the mathematical puzzle in the PoW consensus protocol.

2) CONSORTIUM BLOCKCHAIN

In the case of a consortium blockchain, every node do not have the same privileges when it comes to transaction validation. Only a few nodes are given special permissions to validate transactions. The rest of them may agree, but before the implementation, this small group of nodes must agree.

3) PRIVATE BLOCKCHAIN

The totally private blockchain is a somewhat different from the consortium blockchain. It is organized in a centralized manner. A single entity has the authority to make decisions, and this entity also controls the validation process. The centralized head will ensure that the consensus reached is the one that was mentioned.

Permissionless blockchain refers to the public blockchain system, whilst permissioned blockchain refers to the other two. The characteristics of both are summarized in Table 1. We can infer that private or permissioned blockchain has several advantages over pubic or permissionless.

TABLE 1. Permissionless Vs Permissioned Blockchain [13], [14].

Parameter	Permission-Less	Permissioned
Blockchain Type	Public	Private and Consortium
Number of Nodes	High	Limited to Organization
Type of Nodes	Unknown and Untrusted	Known and Verified
Energy Efficient	Less	High
Computing Speed	Low	High
Decentralization	High	Less
Latency	High	Low
Usecase	Bitcoin	Corda

III. CONSENSUS MECHANISM IN BLOCKCHAIN

The consensus is the process of agreement; all the nodes decide how an agreement is made to append a new block to an existing blockchain. The consensus mechanism is the core of any blockchain-enabled system; a system is as strong and reliable as the consensus protocol that governs it [15]. There exists a wide range of consensus mechanisms by which participating nodes of a blockchain can achieve consensus to append a new block. The various consensus protocols are discussed in detail [16]–[21]. The consensus protocols are broadly categorized on the basis of their working mechanism. There are two types of consensus protocols: i) Proof based and ii) Voting Based. Another way of categorization is on the basis of the type of blockchain used in the system. The permissionless blockchain-primarily utilizes proof-based consensus protocols while the permissioned blockchain uses voting-based protocols. Also, one more type of classification is based on the fault a consensus mechanism can tolerate. The most common faults that may occur in blockchain-enabled systems are double-spending, byzantine faults, and crash faults. Categorization is based on the type of fault tolerance; we have a consensus mechanism that can work even after having byzantine faults and crash faults. In the next section, we will discuss the most prominent protocols and also give a comparative study. Consensus in

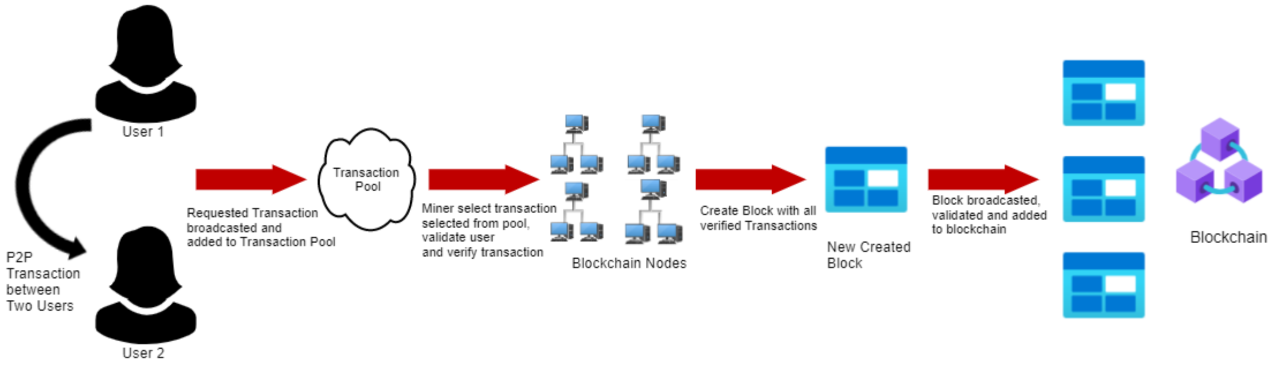
User 1
P2P Transaction between Two Users
User 2
Requested Transaction broadcasted and added to Transaction Pool
Transaction Pool
Miner select transaction selected from pool, validate user and verify transaction
Blockchain Nodes
Create Block with all verified Transactions
New Created Block
Block broadcasted, validated and added to blockchain
Blockchain

FIGURE 3. Working of Blockchain.

the distributed or decentralized networks can be achieved by defining any criteria such as quorum structure, decentralized governance, computing power, byzantine fault tolerance, etc. The public blockchain such as Bitcoin incorporates the concept of "Proof of Work". Proof of Stake (PoS), Proof of Elapsed Time (PoET), Delegated Proof of Stake (DPoS), Proof of Existence (PoE), Proof of Importance, Proof of Storage, and other hybrid proof-based consensus algorithms while the consortium and private blockchain prefer to have voting-based consensus protocols such as PBFT, RAFT, Ripple and Stellar, etc. A proper classification is presented in Fig. 4.

A. PROOF-BASED CONSENSUS PROTOCOLS

1) PROOF OF WORK (POW)

The proof-of-work (PoW) consensus process is at the heart of numerous cryptocurrencies, including bitcoin and Ethereum. It is also known as the Nakamoto consensus protocol, it addresses the problem of poor synchronization in the network. In this mechanism, a hash (a lengthy string of letters) that matches the requested hash for the current block is generated. The widely used proof-of-work consensus algorithm is based on SHA-256. In brief, PoW requires to solve a computationally hard puzzle in order to create new blocks in the Bitcoin blockchain. It will generate a cryptographic hash function that must satisfy a pre-defined condition for the proposed new block. A brief working is shown in Fig.5. It necessitates a lot of processing power and low throughput. The protocol validates the record with the longest transactional history, which is a serious flaw in this sort of protocol. The attacker will have the longest transactional history if they control more than half of the processing power. As a result, their faulty blocks will be the ones that are legitimate. The 51 percent Rule [22] refers to a situation in which a single entity owns more than 51 percent of a blockchain network's computing (hashing) capacity. The entity then prepares transaction records that have been manually verified. Previous payments may not be included in these records, resulting in a double payment.

2) PROOF OF STAKE (POS)

Proof of stake protocols are the second most used method of consensus. It is a type of blockchain consensus method that works by selecting validators based on their holdings of the associated currency. This does not include a race amongst the nodes for adding the next block. The next block is selected depending on the network's proportional ownership. The stake is the amount of money it has in that cryptocurrency. This method avoids the computational costs associated with proof-of-work techniques. This solution, although removing the computational constraints of proof of work, introduces additional issues. This system is reliant on nodes with the largest stake, which renders the blockchain centralized in some way.

3) DELEGATED PROOF OF STAKE (DPOS)

The approach is based on the PoS mechanism. This system is in contrast to PoS, which is democratized directly, implying that all stakeholders have a say to appoint some nodes as delegates and witnesses [23]. In a delegated proof of stake system, stakeholders reach a consensus depending on the amount of stake they have in a cryptocurrency system. According to experts, some of the benefits of delegated proof of stake include scalability and speed, as well as the streamlining of digital transactions. Concerns regarding security and equality emerge, however, since delegated proof of stake tends to concentrate decision-making in the hands of the wealthiest few in a particular cryptocurrency market. Some are concerned that a delegated proof of stake approach may lead to bigger stakeholders creating cartels, which might result in a range of negative market activities, a comparision of PoS and DPoS is mentioned in [24].

4) PROOF OF ELAPSED TIME (POET)

Proof of elapsed time (PoET) [25] is a blockchain network consensus technique that limits resource and energy usage through a random lottery system. Similar to PoW, the miner has to solve a puzzle but with a focus on consumption. Instead of having a competition between miners, the miner is selected

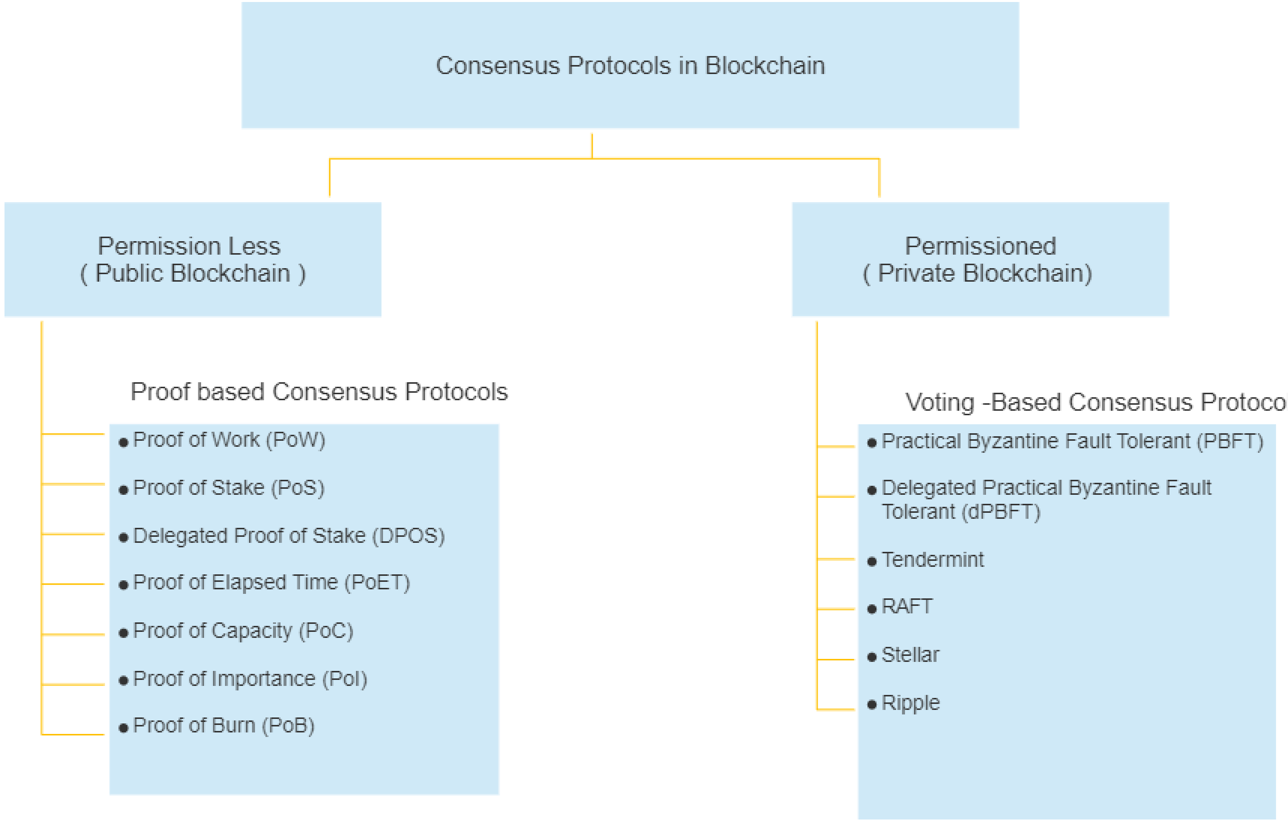
Consensus Protocols in Blockchain
Permission Less
(Public Blockchain)
Permissioned
(Private Blockchain)
Proof based Consensus Protocols
Proof of Work (PoW)
Proof of Stake (PoS)
Delegated Proof of Stake (DPOS)
Proof of Elapsed Time (PoET)
Proof of Capacity (PoC)
Proof of Importance (PoI)
Proof of Burn (PoB)
Voting -Based Consensus Protocols
Practical Byzantine Fault Tolerant (PBFT)
Delegated Practical Byzantine Fault Tolerant (dPBFT)
Tendermint
RAFT
Stellar
Ripple

FIGURE 4. Classification of Consensus Protocols.

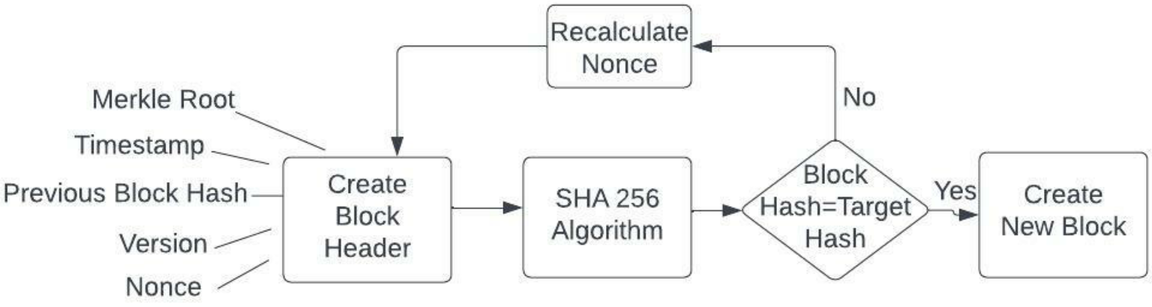

FIGURE 5. Proof of Work (PoW).

on the basis of a random timer. Each network member is assigned a random timer object, and when the first timer expires, that member "wakes up" and becomes the block leader, responsible for creating the next block. The correctness of time is done using the TEE (Trusted Execution Environment) system. PoET was developed by the semiconductor manufacturer Intel as an efficient consensus technique for permissioned blockchain networks. PoET consensus is quick like proof of work that substitutes mining with a randomized timing scheme for network members. PoET is widely used for developing and experimenting with permissioned distributed ledger systems, and it is currently the consensus model of choice for the modular structure of Hyperledger Sawtooth [26].

5) PROOF OF CAPACITY (POC)

Proof of capacity (PoC) [27] is a blockchain consensus mechanism approach that enables network mining devices to identify mining rights and validate transactions by using available hard drive space. The computing power issue in proof-of-work (PoW) as systems need a lot of energy, and proof-of-stake (PoS) systems use a lot of money, both can be addressed by PoC. The more possible solution values that may be saved on a hard drive, the better a miner's odds of matching the required hash value from his list, and thus the greater his chances of earning the mining prize.

6) PROOF OF IMPORTANCE (POI)

Proof of Importance (PoI) takes into account more factors than just nodes' deposits while identifying the next block. For example, the number of transactions that occurred to or from that node is taken into account.

7) PROOF OF BURN (POB)

Proof of Burn is built on the idea of ''burning'' coins, which is defined as transferring coins to an address that cannot be recovered. Miners are given precedence in solving the next block based on how many bitcoins they have burned [28]. This method can be used to create a cryptocurrency.

B. VOTING-BASED CONSENSUS PROTOCOLS

The proof-based consensus protocols normally need high computational power and are less tolerant to faults. Therefore, researchers implemetns the conventional mechanism for consensus in blockchain which are already successfully implemented in distributed systems i.e., voting-based consensus protocols. The voting-based protocols give each participating node a chance to actively participate in the decision of adding a block into the chain. Also, the transactions will be verified by a majority of participating nodes present in the network, which gives more tolerance against faults. Voting-based protocols are normally used by permissioned or private blockchains, however, some of them are used by both private and public blockchains such as stellar and ripple consensus protocols. In the next section, we will discuss various voting-based protocols.

1) PRACTICAL BYZANTINE FAULT TOLERANT (PBFT)

All the participated nodes perform a process of voting to append the next block in the existing blockchain. When the majority of nodes agree to add a block to the chain, the selected block is added to the chain. In PBFT the majority is considered as two-thirds of the total number of participating nodes. PBFT can tolerate one-third of malicious nodes to perform properly. In PBFT the consensus is achieved faster and it is more economical compared to PoW. This is the most appropriate method for private blockchains such as Hyperledger projects. It is not considered good for public blockchain due to less scalability and limited fault tolerance. PBFT has high throughput, low latency, and low computational overhead [29]. The PBFT protocol allows a distributed network to achieve the consensus even if some nodes are malicious. It is implemtned in Hyperledger where, a certain number of nodes must agree for a transaction to be accepted [30]. The transaction details are transmitted to the network's nodes after a Hyperledger transaction is finished. In PBFT the process is divided into number of phases. It has *pre-prepare, prepare* and *commit* phase before making a final consensus decision. A brief working is shown in Fig.6.

2) DELEGATED PRACTICAL BYZANTINE FAULT TOLERANCE (DPBFT)

Delegated Practical Byzantine Fault Tolerance is similar to PBFT however, unlike PBFT each node participation for adding the block is not required here, this makes it more scalable. Here, few nodes are selected as delegates of other nodes and these nodes are responsible for achieving consensus [31].

3) TENDERMINT

Tendermint is a member of the Byzantine Fault Tolerance (BFT) consensus protocol family, which allows for the hosting of arbitrary application states. It's a method of permissioned consensus. In Tendermint [32], nodes have varied voting powers according to their stakes, unlike PBFT where each node has the same voting power. As a result, it can be thought of as a hybrid consensus approach that combines PBFT and PoS.

4) RAFT

Raft [33] is a voting-based consensus mechanism developed to make the Paxos algorithm more intelligible and implementable in real-world systems. The Paxos algorithm [34] attempts to address the Byzantine Generals Difficulty's consistency problem under particular conditions. The efficiency of Raft is comparable to that of Paxos. Raft and Paxos are non-Byzantine fault tolerance algorithms. These protocols are similar to BFT algorithms; however, they can only tolerate up to 50% of nodes being corrupted. Leader election and log replication are the two stages of the RAFT algorithm. The leader is in charge of placing everything in its proper place. When an existing leader fails, a randomized timeout is used to select a new leader for each server. When a leader is chosen, the log replication step begins. The leader receives client log entries and broadcasts transactions in this stage to create its own version of the transaction log. The Corda blockchain [35] is a version that uses Raft as a consensus technique.

5) RIPPLE

Ripple Consensus Protocol: Ripple works as a cryptocurrency and as a digital payment network. It uses a unique consensus mechanism through a network of servers to validate transactions. Its network is private and one needs permission to become a part of it. All the servers or the nodes on the network conduct a poll and decide whether the transaction is valid and authentic. This allows instant confirmations without the presence of any central authority. It is much more decentralized and is more reliable. Ripple cryptocurrency maintains the record of transactions made across various computers but not anyone can become a node on the network. Also, its unique consensus mechanism gives it a slight deviation from being just a private blockchain [36].

6) STELLER

It possesses the characteristics of both types of systems, permissioned and permission-less. The Steller protocol lacks a central gatekeeper to monitor and ensure transaction

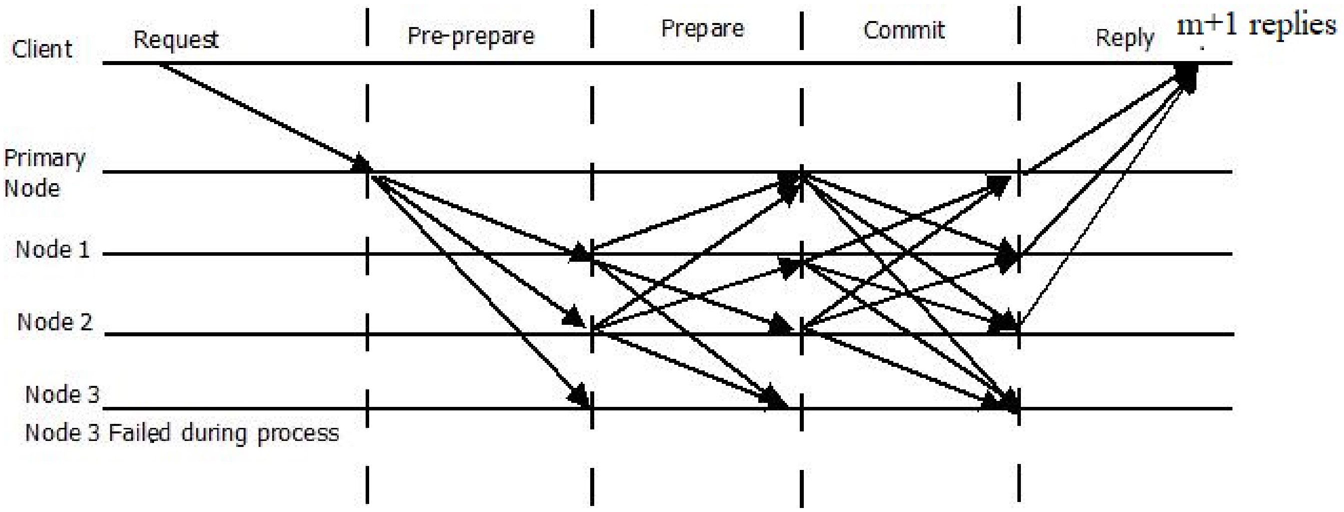

FIGURE 6. Practical Byzantine Fault Tolerance (PBFT) Consensus Protocol.

legitimacy. Nodes that do join the Stellar network, operate on the basis of shared quorum slices, this indicates that when multiple nodes have the same state value, the majority of nodes have the same state value. This serves as a fictitious key for nodes in the validating network. In this case, permission is a privilege granted to a node to participate in transaction validation if it shares the state value with other nodes. There is no central authority to seek permission from, but because the system is founded on trust, you must be trusted by someone who is already trusted for your validation opinions to be valid [37].

IV. FORMAL APPROACH IN CONSENSUS PROTOCOLS

This section discusses the requirement of formal methods in blockchain technology and in consensus algorithms. A brief description of the formal modeling and verification techniques applied in blockchain based system is mentioned in this section. Before diving into the formal methods, we discuss the importance of formal methods in blockchain based system and how consensus mechanism and formal methods help us to achieve trustworthiness in these system.

A. TRUST IN BLOCKCHAIN

It is essential to ensure the proper working of smart contracts to create trust in blockchain-based systems. The blockchain relies on cryptographic rules and mathematics to bring confidence in the system. The trust relies on the proper operation and governance of the underlying protocols. The consensus protocols are useful in building the trust in the system [43]. Developing a correct and reliable system, has been a continual problem. Various techniques are implemented to achieve the same. An effective way to solve this problem is formal methods. The formalization and verification of blockchain's safety and security protocols is becoming increasingly important as blockchain has its application in various fields such as cryptocurrency, security solutions [44], sustainable energy system [45], supply-chain management [46], and other sectors like healthcare, agriculture and digital distribution system [47]–[49]. Formal verification can be used to model and verify properties like liveness, safety, and fault tolerance, together with blockchain consensus-specific properties. Aforementioned, consensus protocols are used to build trusted blockchain based system. Formal verification of the consensus algorithms provides credibility in blockchain enterprises that aids investors in making decisions.

1) FORMAL METHODS

The term formal techniques [50] refers to the application of methods for mathematical modelling, computation, and prediction in specification, design, analysis and construction of hardware and software systems. These techniques are distinguished by their well-defined syntax and semantics. The problem of assuring the correctness of complex software systems can only be solved via formal methods [51], [52]. By conducting mathematical formalization of the security needs, specifications, and operational environment, one can analyze the likelihood of an attack on the specification of the protocol, products, or system. Formal specifications methodologies provide a detailed and unambiguous description of system properties, which are useful for avoiding misconceptions and subsequent system verification. The process of describing a system and its desired attributes is known as specification. Formal specification employs a syntax and semantics that are mathematically defined. The specification language limits what may be expressed, formal techniques that can be used, and what can be automated to some degree. Every specification language is based on a well-known mathematical or logic theory and has a formal semantics. Set theory, process algebra, first-order logic, state transition systems, higher-order logic, temporal logic, are all used to create specification languages. Specification

TABLE 2. Comparative Study of Consensus Protocols on Various Parameters [38]–[42].

Consensus Methods	Accessibility	Decentralization	Scalability	Throughput	Latency	Adversary tolerance	Computing Overhead	Network Overhead
PoW	Public, PL	High	High	Low	High	<25%Computing Power	High	Low
PoS	Public, PL	High	High	Low	Medium	<51%Stakes Power	Medium	Low
DPoS	Public, PL	Medium	High	High	Medium	<51%Validators	Medium	NA
PoET	Public, PL,P	High	High	High	Low	NA	Low	Low
PoC	Public, PL	High	High	Low	High	NA	Low	Low
PoI	Public, PL	High	High	High	Medium	<51%Importance	Low	Low
PoB	Public, PL	High	High	Low	High	<25%Computing Power	Medium	Low
PBFT	Private, P	Medium	Low	High	Low	<33%Faulty Replica	Low	High
dPBFT	Private, PL	Meduim	High	High	Medium	<33%Computing Power	Low	High
RAFT	Private, P	Medium	High	High	Low	<50%Crash Fault	Low	NA
Steller	Public, PL	High	High	High	Meduim	<Variable	Low	Meduim
Ripple	Public, PL	High	High	High	Meduim	<20%Faulty UNL	Low	Meduim

process uncovers design flaws, inconsistencies, ambiguities, and incompleteness. Once the system is formally described by its specifications, it must be verified. Formal verification is the process of checking the correctness of a protocol's, product's, or system's specification using formal methods such as automated axiomatic theorem proving or model checking. A system is expressed as a formal model in model checking approach, which describes its behavioral features in formal language. The act of generating a mathematical proof for a mathematical assertion to be true is known as theorem proving. The system and its properties are expressed in terms of mathematical logic in theorem proving techniques.

B. NEED OF FORMAL METHODS IN BLOCKCHAIN CONSENSUS MECHANISM

The blockchain is a disruptive technology with far-reaching implications as a decentralised and distributed consensus framework for maintaining and securing a shared ledger. The major cloud platform providers such as Microsoft, IBM, Amazon, SAP, and Oracle are launching Blockchain-as-a-Service (BaaS). According to a recent Gartner analysis, the corporate value-add of blockchain technology might reach $3.1 trillion by 2030 [53]. As the application of blockchain in different fields is increasing, the formalization and verification of blockchain's safety and security features is becoming essential. Also, due to the dynamics of the cryptocurrency ecosystem, formal methods should be gradually included into cryptocurrency software development [54]. As blockchain enabled system are immutable hence, transactions are irreversible, any defects in smart contract code can have disastrous repercussions, and smart contract vulnerabilities may have result in a huge loss in creating trust in the underlying blockchain technology. The challenges and possible attacks on blockchain have already been a research topic for researchers [55]–[57]. The infamous DAO bug [58] resulted in the loss of over $60 million in Ether, and the Parity Wallet issue led in the permanent locking of 169 million USD in Ether [59]. The only way to fix these problems was to hard-fork the blockchain and restore one of the forks to its previous configuration. However, this solution is ineffective because it undermines blockchain's key qualities of immutability, decentralised trust, and self-governance. Smart contract programmers have no choice but to write correct code from the beginning. Motivated by the effects of bugs in smart contract code several ways have been investigated to mitigate these assaults and avoid breaches. These ways incudes proper documentation of vulnerabilities or model validation using formal verification. Various formal approaches are used to improve the security and overall trustworthiness of the blockchain system for end users. Miller A. *et.al.* [60] evaluated how defects and errors in growing technology can result in significant financial loss, which could be viewed as an opportunity for formal methods to be used in this new bitcoin model or cryptocurrency. Systematic procedures and formal approaches for system development can result in bug-free, minimal defects, accurate, guaranteed-correct, certified software [61]. A number of important software systems have been designed using formal methods. In blockchain based system if level of abstraction are applied, we will have two layers where we can apply formal methods. First one is the outer layer or the system-level where we study the interaction and the overall behavior of the system with external entities or users, it doesn't include technical and execution details of its implementation while the second level is the innermost layer where implementation details are written, it is also termed as program-level. The program-level includes the source code and executional details of underlying protocols. In consensus protocols, low-level details of execution are considered therefore we can say that consensus protocol is the executional aspect of blockchain based system and hence formal techniques on program-level are applied.

V. FORMAL TECHNIQUES APPLIED IN BLOCKCHAIN

The formal techniques which can be utilized to perform formal modeling are process algebra, state-transition model, and set-based methods and later on formal verification techniques can be used to verify these models. In this section we will discuss these broadly classified formal techniques and how are they used in blockchains.

A. PROCESS-ALGEBRA

Process algebra [62], is a collection of mathematical techniques for modelling the behaviour of distributed or parallel systems as interacting concurrent processes. There are a variety of techniques for obtaining a rigorous mathematical understanding of the semantics of syntactically accurate process, process algebra is one of them. It comes with a set of constructors and equational axioms for system descriptions, as well as an operational semantics that defines the evolution of systems in terms of labelled transitions [63]. Some of the process algebra which can be applied in blockchain system are discussed in this section.

1) PI-CALCULUS

Pi-calculus or $\pi-$calculusis a small but expressive language with few terms. It can be used to encode functional programs. The $\pi-$calculus and applications have proven successful in reasoning about cryptographic protocols and it is also applied in formal analysis of consensus protocols in blockchain [64]. In [65] a process calculus referred as natural calculus for consensus protocol is proposed. The proposed calculus is used as specification language for describing consensus protocols.

2) COMMUNICATING SEQUENTIAL PROCESSES(CSP)

CSP [66], is a language that analyses communication between systems using math and logic. Concurrent systems communicate by passing messages, while sequential processes communicate with one another in CSP. This communication is logical and algebraic. CSP was initially developed by Tony Hoare in 1978. CSP is a programming language that can be used to examine software, computer systems, and programming languages. CSP, like process algebra, breaks down processes that occurs at the same time and interact with one another, describing the interactions with algebraic equations and logic. It can be used to check synchronization in concurrent process for blockchain enabled system [67].

3) BitML

It is a smart contract high-level language with a computationally sound embedding in Bitcoin and a sound and full verification technique of important trace features. Many smart contracts can be expressed in BitML and executed by appending appropriate transactions to the Bitcoin blockchain. Bartoletti, M and Roberto Z. [68] created a toolchain for creating, validating, and implementing BitML contracts on Bitcoin. The suggested tool can be used as a security analyzer and checking arbitrary LTL properties.

B. STATE TRANSITION

A transition system is a notion in theoretical computer science that is used to investigate computation. It's a term used to explain how separate systems might behave. It is made up of states and transitions between states that can be labelled with labels from a list; the same label can appear on many transitions. A brief introduction of various state-transition techniques which can be used for modeling the system is given here.

1) PERTI NETS

A Petri Net is a model for describing systems that uses a bipartite graph with two types of nodes: locations and transitions. Petri Nets are also known as Place Transition nets (P/T nets). Directed arcs serve as node-to-node connections. Arcs can be divided into pre-arcs that lead into a transition and post-arcs that lead out of a transition. One of the benefits of using Petri Nets is that an algebraic formalism can be used to describe them. In blockchain, Pinna *et.al.* [69], examined two items: addresses and transactions and provides a set theory explanation of both elements, in order to achieve the Petri Net algebraic representation.

2) TIME-AUTOMATA (TA)

The finite automata model is associated with clock variables. The formal modeling is done by writing simple constraints over clocks and states in the timed automata (TA). The timed automata model has been successfully employed for real-time systems' verification and is the foundation of various model-checking tools. In [70], a framework for modelling Bitcoin contracts using timed automata is proposed and the proposed model is then confirmed using the Uppaal model checker.

3) MARKOV DECISION PROCESSES (MDP)

A Markov Decision Process (MDP) is fundamentally a mathematical framework used for decision making in transition model, the transition model is described by Markov model. The MDP can be used to evaluate the performance of blockchain systems. In [71], the problem of addition of block in blockchain is analyzed. It is observed that as user has to wait to get his block appended in the chain. In this study Markov queue model is used to reduce the waiting time for the blocks and to get transaction confirmation by improving the approval rate of transactions. Also, in order to identify an optimal selfish mining strategy in blockchain systems, several studies utilized MDP as a mathematical modeling framework [72], [73] are conducted by researchers.

C. SET-BASED

Frameworks such as event-B [74] and TLA+ [75] based on set-theory and logic are used to give program level models i.e., to model executional part of the system. It analyzed how the system work has been performed. Zhu *et.al.* [76] provide a formal verification of solidity written smart contract in event-B. The objective is to verify and validate the safety, correctness and functional accuracy of smart contracts with their specified behavior. In [77], Lahbib *et.al.* translated smart contract into event-B models and verified by using RODIN platform. The cryptographic protocols and consensus protocols used in blockchain may be verified using TLA+ framework [78].

For formally writing or modeling the blockchain enabled system following techniques can also be used along with the above mentioned techniques. (i).**Control-Flow Graphs (CFG)**: The complete flow of the execution is represented as the path of the graphs. The simplest unit of control flow in a program is a basic block. A basic block is a sequence of operations that always execute together, unless an operation raises an exception. Directed edges are used to represent jumps in the control flow. (ii).**Abstract Syntax Tree (AST)**: The blockchain system can be represented as a hierarchical tree structures. Blockchains is described as a composition of abstract data types all together with a hierarchy of consistency criteria that captures the eventual convergence process in blockchain systems. (iii).**Linear Temporal Logic (LTL)**: It is used for System-level Specification. It's made up of a finite number of propositional variables, the logical operators and the temporal model operators. The critical aspects of a consensus protocol include the safety, fault tolerance, leader trust and validator trust. These all are formally stated using Linear Temporal Logic (LTL) to protect against a variety of security threats. (iv).**Computation Tree Logic (CTL)**: Computation tree logic (CTL) is a branching-time logic, it is a tree like structure where future is not predetermined, there are multiple pathways in the future, any of which could be a real-world path. It is often employed in formal verification of software or hardware system. In software applications, an application referred as model checker to assess whether a given system has safety or liveness properties is used. CTL can specify that all possible program executions to avoid some undesired condition (e.g., dividing a number by zero). A model checker verifies the safety property by exploring all potential transitions out of program states that fulfil the starting condition and ensuring that all such executions satisfy the property. Computation tree logic belongs to the same family of temporal logics as linear temporal logic (LTL). A system is modeled or formally specified by writing the specification and a formally specified system is verified. The verification of a formally modeled system majorly depends on the technique used in writing the specification or formal model technique used for modelling the system. Along with two well-known techniques i.e., Model checking and Theorem Proving, we have symbolic notation and program verification techniques used for verification of the modeled system. Temporal properties written in TLA+, are verified by a model checker tools. System written using CFG, are verified by symbolic execution techniques. In comparison to other verification techniques, theorem proving is more complex, generally written in Hoare logics. Therefore, model-checking technique is used for verification [79], [80].

D. FORMAL APPROACH IN CONSENSUS PROTOCOLS

We have discussed different types of the consensus protocols in section III, proof-based consensus and voting-based consensus protocols. We have seen in comparative study of these two types of consensus protocols that voting based protocols for newly developed private and consortium blockchain has advantages over proof-based consensus protocols. The voting-based mechanism need less computational power and has high throughput, also they are more fault tolerant. The process of achieving consensus is more algorithmically practical by voting-based protocols. The application of formal methods is majorly conducted in voting-based consensus protocols due to their numerous advantages. We found very few articles on formal modeling and verification of consensus protocol, as the technology is emerging. In the beginning we studied the PAXOS [34] algorithm developed by Lamport as it is the first ever algorithm for reaching consensus in asynchronous system. Also, it is the foundation of many consensus protocols used in blockchain. It is highly efficient and fault tolerant algorithm. However, initially it failed to impress the other researchers. Later, PAXOS re-written by Lamport, that is widely accepted in its various forms and implemented in system like Google file System [81]. For better understanding of PAXOS, a formal presentation is given in [82], a formal framework in time automata is presented, specifically in Clock General Timed Automaton (Clock GTA) model. It provides a systematic time-based description of the protocol. Later, in [83], formal modeling of PAXOS is presented in finite state automata with specification written in Promela language. It is a high-level language in which guards are written in a non-ambiguous executable semantics. The model further verified by SPIN model checker tool. Other well known voting-based consensus protocol used in industry is Raft. Raft can be termed as the extension of PAXOS, it is widely used in private blockchain. It is more understandable and practical than PAXOS. It is used in Zookeeper, Facebook, and Google. Raft consensus protocol has great research scope in formal modeling. With the release of Raft its partial formal specification is presented in TLA+ [84]. In [85], new functionalities are added in already presented formal specification of Raft to provide safety property in leader election phase of Raft algorithm. This work ensures that there must be at-most one leader per term in Raft protocol. The first formally verified implementation of Raft algorithm, a model for the primary safety property i.e., state machine safety is presented in [86]. Linearizability of protocol along with basic state machine replication property, which state that each replicated state machine has the same sequence of events and execute same commands in same order and other properties regarding the leader election is also formally verified. In [87], the formal modeling of Raft in LNT process algebra is presented which is verified using CADP [88], model checking techniques. The CADP tool is used to represent a labelled transition system. Here, author identified state change issue from candidate to follower in formal TLA+ specification of Raft. In [89], interactive preserving abstraction (IPA) framework is used for verification of RAFT using TLA+ language. In Raft protocol leader election and log replication, two phase has been identified. When the elected leader fails, transfer of leadership or the re-election of leader may cause the cluster unavailable. In [90], this situation has been formally modeled using TLA+ language and verified using TLC model checker. Model checking has been a powerful tool to verify complex system. However, it has been rarely applied to consensus algorithm in asynchronous distributed system due to huge number of states in these consensus protocols. Also, it is not possible to verify the protocol in every possible state. A formal verification technique is required which is simple to apply in these consensus protocols. In [91], a computational model which provide high level abstraction developed for consensus algorithm based on Heard-Of-model (HO model) and provide complete verification of consensus protocols for asynchronous system. Formal verification of BFT is performed in [92] using ByMC model checker. The quorum based BFT consensus protocol Steller is modeled in [93]. The stated methodology for formally verifying the safety and liveness of Steller protocol is proposed. The protocol is modeled in first order logic and Isabelle/HOL in combination with Ivy used for verification. To ensure the veracity of consensus processes, those must be formalized and verified. Model-checking is a well-known formal verification methodology based on formal methods. Model-checking is the process of assessing whether the formal model of a system meets the requirements. The model checking technique is beneficial for revealing underlying flaws that testing and simulation techniques are many time unable to detect.

VI. CONCLUSION

Blockchain has lately been one of the technologies that is attracting a lot of interest. Being a relatively new research area, a standard or best practice for formal verification of blockchain and smart contracts has not yet been established. A formal proof of the consensus algorithms gives credibility to companies using blockchain approaches and help investors to make decisions. We have discussed various formal methods that can be applied to blockchain enabled system. We have emphasize that writing formal specification for consensus protocols and for blockchain-enabled system is important to develop a correct system. After rigourous study it has been observed that for verification of the blockchain-enabled system, model checking is the most adopted technique. The integration of formal methods in this rapidly adopted technology will gain more importance in the near future and has great scope for researchers.

Conflict of Interest All authors declare that they have no conflicts of interest.

REFERENCES

[1] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," in *Proc. Conf. Theory Appl. Cryptogr.* Berlin, Germany: Springer, Aug. 1990, pp. 437–455.
[2] S. Nakamoto and A. Bitcoin. *A Peer-to-Peer Electronic Cash System*. Accessed: Mar. 4, 2022. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[3] W. Chen, Z. Xu, S. Shi, Y. Zhao, and J. Zhao, "A survey of blockchain applications in different domains," in *Proc. Int. Conf. Blockchain Technol. Appl. (ICBTA)*, 2018, pp. 17–21.
[4] J. Gantz and D. Reinsel, "Extracting value from chaos," IDC iView, Feamingham, MA, USA, Tech. Rep. IDC 1142, 2011.
[5] M. Dabbagh, M. Sookhak, and N. S. Safa, "The evolution of blockchain: A bibliometric study," *IEEE Access*, vol. 7, pp. 19212–19221, 2019.
[6] V. Buterin, "A next-generation smart contract and decentralized application platform," White Paper 3, 2014, no. 37.
[7] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-enabled smart contracts: Architecture, applications, and future trends," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 11, pp. 2266–2277, Nov. 2019.
[8] N. Szabo, "Smart contracts," Virtual School, Tech. Rep., 1994. Accessed: Mar. 17, 2022. [Online]. Available: http://szabo.best.vwh.net/smart.contracts.html
[9] R. Beck, "Beyond bitcoin: The rise of blockchain world," *Computer*, vol. 51, no. 2, pp. 54–58, Feb. 2018.
[10] I. Bashir, *Mastering Blockchain*. Birmingham, U.K.: Packt, 2017.
[11] B. Singhal, G. Dhameja, and P. S. Panda, *Beginning Blockchain: A Beginner's Guide to Building Blockchain Solutions*. New York, NY, USA: Apress, Jul. 2018.
[12] A. Vacca, A. Di Sorbo, C. A. Visaggio, and G. Canfora, "A systematic literature review of blockchain and smart contract development: Techniques, tools, and open challenges," *J. Syst. Softw.*, vol. 174, Apr. 2021, Art. no. 110891.
[13] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "BLOCKBENCH: A framework for analyzing private blockchains," in *Proc. ACM Int. Conf. Manage. Data*, May 2017, pp. 1085–1100.
[14] S. Pongnumkul, C. Siripanpornchana, and S. Thajchayapong, "Performance analysis of private blockchain platforms in varying workloads," in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2017, pp. 1–6.
[15] K. L. Jones, "Blockchain: Building consensus and trust across the space sector," in *Proc. 35th Space Symp., Tech. Track*, Colorado Springs, CO, USA, Apr. 2019, vol. 4, no. 8, p. 19.
[16] G.-T. Nguyen and K. Kim, "A survey about consensus algorithms used in blockchain," *J. Inf. Process. Syst.*, vol. 14, no. 1, pp. 101–128, Jan. 2018.
[17] C. Cachin and M. Vukolić, "Blockchain consensus protocols in the wild," 2017, *arXiv:1707.01873*.
[18] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," in *Proc. 4th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Jan. 2017, pp. 1–5.
[19] M. S. Ferdous, M. J. M. Chowdhury, M. A. Hoque, and A. Colman, "Blockchain consensus algorithms: A survey," 2020, *arXiv:2001.07091*.
[20] S. Zhang and J.-H. Lee, "Analysis of the main consensus protocols of blockchain," *ICT Exp.*, vol. 6, no. 2, pp. 93–97, Jun. 2020.
[21] M. S. Ferdous, M. J. M. Chowdhury, M. A. Hoque, and A. Colman, "Blockchain consensus algorithms: A survey," 2020, *arXiv:2001.07091*.
[22] S. Sayeed and H. Marco-Gisbert, "Assessing blockchain consensus and security mechanisms against the 51% attack," *Appl. Sci.*, vol. 9, no. 9, p. 1788, Apr. 2019.
[23] D. Larimer, "Delegated proof-of-stake (DPoS)," Bitshare White Paper 81, 2014.
[24] S. M. S. Saad, R. Z. R. M. Radzi, and S. H. Othman, "Comparative analysis of the blockchain consensus algorithm between proof of stake and delegated proof of stake," in *Proc. Int. Conf. Data Sci. Its Appl. (ICoDSA)*, Oct. 2021, pp. 175–180.
[25] J. Frankenfield, "Proof of elapsed time (PoET)(Cryptocurrency)," Tech. Rep., 2020. Accessed: Feb. 3, 2022. [Online]. Available: https://www.investopedia.com/terms/p/proof-elapsed-time-cryptocurrency.asp
[26] *Hyperledger*. Accessed: Apr. 16, 2018. [Online]. Available: https://www.hyperledger.org
[27] K. Sharma and D. Jain, "Consensus algorithms in blockchain technology: A survey," in *Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2019, pp. 1–7.
[28] K. Karantias, A. Kiayias, and D. Zindros, "Proof-of-burn," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Cham, Switzerland: Springer, Feb. 2020, pp. 523–540.
[29] X. Zheng and W. Feng, "Research on practical byzantine fault tolerant consensus algorithm based on blockchain," *J. Phys., Conf. Ser.*, vol. 1802, no. 3, Mar. 2021, Art. no. 032022.

[30] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. de Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, ''Hyperledger fabric: A distributed operating system for permissioned blockchains,'' in *Proc. Eur. Conf. Comput. Syst.*, Apr. 2018, pp. 1–15.

[31] Y. Zhan, B. Wang, R. Lu, and Y. Yu, ''DRBFT: Delegated randomization byzantine fault tolerance consensus protocol for blockchains,'' *Inf. Sci.*, vol. 559, pp. 8–21, Jun. 2021.

[32] *Tendermint*. Accessed: Mar. 7, 2022. [Online]. Available: https://tendermint.com

[33] O. Diego and J. Ousterhout, ''In search of an understandable consensus algorithm,'' in *Proc. USENIX Annu. Tech. Conf. (USENIXATC)*, 2014, pp. 305–319.

[34] L. Lamport, ''Paxos made simple,'' *ACM SIGACT News Distrib. Comput. Column*, vol. 32, no. 4, pp. 51–58, Dec. 2001.

[35] M. Hearn and R. G. Brown, ''Corda: A distributed ledger,'' Corda Tech., New York, NY, USA, White Paper 2016, 2016. Accessed: Mar. 23, 2022. [Online]. Available: https://www.corda.net/wp-content/uploads/2019/08/corda-technical-whitepaper-August-29-2019.pdf

[36] D. Schwartz, N. Youngs, and A. Britto, ''The ripple protocol consensus algorithm,'' Ripple Labs, San Francisco, CA, USA, White Paper 5.8, 2014, p. 151. Accessed: Apr. 8, 2022. [Online]. Available: https://ripple.com/files/ripple_consensus_whitepaper.pdf

[37] D. Mazieres, ''The stellar consensus protocol: A federated model for internet-level consensus,'' *Stellar Develop. Found.*, vol. 32, pp. 1–45, Apr. 2015.

[38] V. Sharma and N. Lal, ''A novel comparison of consensus algorithms in blockchain,'' *Adv. Appl. Math. Sci.*, vol. 20, no. 1, pp. 1–3, Nov. 2020.

[39] X. Fu, H. Wang, and P. Shi, ''A survey of blockchain consensus algorithms: Mechanism, design and applications,'' *Sci. China Inf. Sci.*, vol. 64, no. 2, Nov. 2020.

[40] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, ''A survey on consensus mechanisms and mining strategy management in blockchain networks,'' *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[41] N. Chalaemwongwan and W. Kurutach, ''State of the art and challenges facing consensus protocols on blockchain,'' in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2018, pp. 957–962.

[42] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, ''Blockchain challenges and opportunities: A survey,'' *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[43] P. De Filippi, M. Mannan, and W. Reijers, ''Blockchain as a confidence machine: The problem of trust and challenges of governance,'' *Technol. Soc.*, vol. 62, Aug. 2020, Art. no. 101284.

[44] F. Dai, Y. Shi, N. Meng, L. Wei, and Z. Ye, ''From bitcoin to cybersecurity: A comparative study of blockchain application and security issues,'' in *Proc. 4th Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2017, pp. 975–979.

[45] J. Wu and N. Tran, ''Application of blockchain technology in sustainable energy systems: An overview,'' *Sustainability*, vol. 10, no. 9, p. 3067, Aug. 2018.

[46] S. Abeyratne and R. Monfared, ''Blockchain ready manufacturing supply chain using distributed ledger,'' *Int. J. Res. Eng. Technol.*, vol. 5, no. 9, pp. 1–10, Sep. 2016.

[47] O. Ali, A. Jaradat, A. Kulakli, and A. Abuhalimeh, ''A comparative study: Blockchain technology utilization benefits, challenges and functionalities,'' *IEEE Access*, vol. 9, pp. 12730–12749, 2021.

[48] J. Kishigami, S. Fujimura, H. Watanabe, A. Nakadaira, and A. Akutsu, ''The blockchain-based digital content distribution system,'' in *Proc. IEEE 5th Int. Conf. Big Data Cloud Comput.*, Aug. 2015, pp. 187–190.

[49] Y. Guo and C. Liang, ''Blockchain application and outlook in the banking industry,'' *Financial Innov.*, vol. 2, no. 1, pp. 1–2, Dec. 2016.

[50] J. Woodcock, P. G. Larsen, J. Bicarregui, and J. Fitzgerald, ''Formal methods: Practice and experience,'' *ACM Comput. Surv.*, vol. 41, no. 4, pp. 1–36, 2009.

[51] E. M. Clarke and J. M. Wing, ''Formal methods: State of the art and future directions,'' *ACM Comput. Surv.*, vol. 28, no. 4, pp. 626–643, 1996.

[52] J. P. Bowen and M. Hinchey, ''Ten commandments of formal methods... Ten years on,'' in *Conquering Complexity*. London, U.K.: Springer, 2012, pp. 237–251.

[53] ComputerWorld. *Blockchain to Generate More Than in Revenue by 2023*. Accessed: Mar. 14, 2022. [Online]. Available: https://www.computerworld.com/article/3237465/enterprise-applications/blockchain-to-generate-more than-106b-in-revenue-by-2023.html

[54] Matsuo SI., ''How formal analysis and verification add security to blockchain-based systems,'' in *Proc. Formal Methods Comput. Aided Design (FMCAD)*, Oct. 2017, pp. 1–4.

[55] S. Aggarwal and N. Kumar, ''Attacks on blockchain,'' in *Advances in Computers*, vol. 121. Amsterdam, The Netherlands: Elsevier, Jan. 2021, pp. 399–410.

[56] S. Smetanin, A. Ometov, M. Komarov, P. Masek, and Y. Koucheryavy, ''Blockchain evaluation approaches: State-of-the-art and future perspective,'' *Sensors*, vol. 20, no. 12, p. 3358, Jun. 2020.

[57] W. Zou, D. Lo, P. S. Kochhar, X.-B.-D. Le, X. Xia, Y. Feng, Z. Chen, and B. Xu, ''Smart contract development: Challenges and opportunities,'' *IEEE Trans. Softw. Eng.*, vol. 47, no. 10, pp. 2084–2106, Oct. 2019.

[58] M. I. Mehar, C. Shier, A. Giambattista, E. Gong, G. Fletcher, R. Sanayhie, H. M. Kim, and M. Laskowski, ''Understanding a revolutionary and flawed grand experiment in blockchain: The DAO attack,'' *J. Cases Inf. Technol.*, vol. 21, no. 1, pp. 19–32, Jan. 2019.

[59] P. Praitheeshan, L. Pan, J. Yu, J. Liu, and R. Doss, ''Security analysis methods on ethereum smart contract vulnerabilities: A survey,'' 2019, *arXiv:1908.08605*.

[60] A. Miller, Z. Cai, and S. Jha, ''Smart contracts and opportunities for formal methods,'' in *Proc. Int. Symp. Leveraging Appl. Formal Methods*. Cham, Switzerland: Springer, Nov. 2018, pp. 280–299.

[61] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, ''Systematic mapping studies in software engineering,'' in *Proc. Electron. Workshops Comput.*, Jun. 2008, pp. 1–10.

[62] W. Fokkink, *Introduction to Process Algebra*. Cham, Switzerland: Springer, Dec. 1999.

[63] H. Hermanns, U. Herzog, and J. P. Katoen, ''Process algebra for performance evaluation,'' *Theor. Comput. Sci.*, vol. 274, nos. 1–2, pp. 43–87, Mar. 2002.

[64] P. Tolmach, ''A survey of smart contract formal specification and verification,'' *ACM Comput. Surv.*, vol. 54, no. 7, pp. 1–38, 2021.

[65] W. Jeltsch, ''A process calculus for formally verifying blockchain consensus protocols,'' in *Declarative Programming and Knowledge Management*. Cham, Switzerland: Springer, Sep. 2019, pp. 24–39.

[66] S. D. Brookes and A. W. Roscoe, ''CSP: A practical process algebra,'' in *Theories of Programming: The Life and Works of Tony Hoare*. New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 187–222. Accessed: Feb. 20, 2022. [Online]. Available: https://doi.org/10.1145/3477355.3477365

[67] A. Altarawneh, F. Sun, R. R. Brooks, O. Hambolu, L. Yu, and A. Skjellum, ''Availability analysis of a permissioned blockchain with a lightweight consensus protocol,'' *Comput. Secur.*, vol. 102, Mar. 2021, Art. no. 102098.

[68] M. Bartoletti and R. Zunino, ''BitML: A calculus for bitcoin smart contracts,'' in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 83–100.

[69] A. Pinna, ''A Petri nets model for blockchain analysis,'' *Comput. J.*, vol. 61, no. 9, pp. 1374–1388, 2018.

[70] M. Andrychowicz, ''Modeling bitcoin contracts by timed automata,'' in *Proc. Int. Conf. Formal Modeling Anal. Timed Syst.* Cham, Switzerland: Springer, 2014, pp. 7–22.

[71] R. Srivastava, ''Mathematical assessment of blocks acceptance in blockchain using Markov model,'' *Int. J. Blockchains Cryptocurrencies*, vol. 1, no. 1, pp. 42–53, 2019.

[72] J. Niu and C. Feng, ''Selfish mining in ethereum,'' 2019, *arXiv:1901.04620*.

[73] C. Grunspan and R. Pérez-Marco, ''On profitability of selfish mining,'' 2018, *arXiv:1805.08281*.

[74] J. R. Abrial, *Modeling in Event-B: System and Software Engineering*. Cambridge, U.K.: Cambridge Univ. Press, May 2010.

[75] L. Lamport, *Specifying Systems: The TLA+ language and Tools for Hardware and Software Engineers*. Reading, MA, USA: Addison-Wesley Longman Publishing, 2002. Accessed: Mar. 6, 2022. [Online]. Available: https://dl.acm.org/doi/10.5555/579617

[76] J. Zhu, K. Hu, M. Filali, J.-P. Bodeveix, and J.-P. Talpin, ''Formal verification of solidity contracts in event-B,'' 2020, *arXiv:2005.01261*.

[77] A. Lahbib, ''An event-B based approach for formal modelling and verification of smart contracts,'' in *Advanced Information Networking and Applications*. Cham, Switzerland: Springer, 2020.

[78] V. Kukharenko, ''Verification of HotStuff BFT consensus protocol with TLA+/TLC in an industrial setting,'' in *Proc. Comput. Sci. On-Line Conf.* Cham, Switzerland: Springer, 2021, pp. 77–95.

[79] M. Almakhour, L. Sliman, A. E. Samhat, and A. Mellouk, ''Verification of smart contracts: A survey,'' *Pervas. Mobile Comput.*, vol. 67, Sep. 2020, Art. no. 101227.

[80] Z. Nehai, P.-Y. Piriou, and F. Daumas, ''Model-checking of smart contracts,'' in *Proc. IEEE Int. Conf. Internet Things (iThings), IEEE Green Comput. Commun. (GreenCom), IEEE Cyber, Phys. Social Comput. (CPSCom), IEEE Smart Data (SmartData)*, Jul. 2018, pp. 980–987.

[81] M. Burrows, ''The chubby lock service for loosely-coupled distributed systems,'' in *Proc. 7th Symp. Operating Syst. Design Implement.*, 2006, pp. 335–350.

[82] R. De Prisco, B. Lampson, and N. Lynch, ''Revisiting the Paxos algorithm,'' *Theor. Comput. Sci.*, vol. 243, nos. 1–2, pp. 35–91, 2000.

[83] G. Delzanno, M. Tatarek, and R. Traverso, ''Model checking Paxos in spin,'' 2014, *arXiv:1408.5962*.

[84] D. Ongaro, *Consensus: Bridging Theory and Practice*. Stanford, CA, USA: Stanford Univ., 2014.

[85] B. Amos and Z. Huanchen. (2015). *15–812 Term Paper: Specifying and Proving Cluster Membership for the Raft Distributed Consensus Algorithm*. Accessed: Mar. 22, 2022. [Online]. Available: https://www.cs.cmu.edu/ aplatzer/course/pls15/projects/bamos.pdf

[86] D. Woos, J. R. Wilcox, S. Anton, Z. Tatlock, M. D. Ernst, and T. Anderson, ''Planning for change in a formal verification of the raft consensus protocol,'' in *Proc. 5th ACM SIGPLAN Conf. Certified Programs Proofs*, Jan. 2016, pp. 154–165.

[87] H. Evrard, ''Modeling the raft distributed consensus protocol in LNT,'' 2020, *arXiv:2004.13284*.

[88] H. Garavel, F. Lang, R. Mateescu, and W. Serwe, ''CADP 2011: A toolbox for the construction and analysis of distributed processes,'' *Int. J. Softw. Tools Technol. Transf.*, vol. 15, no. 2, pp. 89–107, Apr. 2013, doi: 10.1007/s10009-012-0244-z.

[89] X. Gu, W. Cao, Y. Zhu, X. Song, Y. Huang, and X. Ma, ''Compositional model checking of consensus protocols specified in TLA+ via interaction-preserving abstraction,'' 2022, *arXiv:2202.11385*.

[90] G. Yu, L. Hua, L. Yuanping, L. Bowei, W. Xianrong, and R. Hongwei, ''Using TLA+ to specify leader election of raft algorithm with consideration of leadership transfer in multiple controllers,'' in *Proc. IEEE 19th Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Jul. 2019, pp. 219–226.

[91] B. Charron-Bost and S. Merz, ''Formal verification of a consensus algorithm in the heard-of model,'' *Int. J. Softw. Informat.*, vol. 3, nos. 2–3, pp. 273–303, 2009.

[92] P. Tholoniat and V. Gramoli, ''Formal verification of blockchain byzantine fault tolerance,'' 2019, *arXiv:1909.07453*.

[93] G. Losa and M. Dodds, ''On the formal verification of the Stellar consensus protocol,'' in *Proc. 2nd Workshop Formal Methods Blockchains (FMBC)*, 2020, pp. 9:1–9:9.

SUDHANI VERMA received the M.Tech. degree from MANIT, Bhopal. She is currently pursuing the Ph.D. degree from the Department of Computer Science and Engineering, Institute of Engineering and Technology, Lucknow. Her research interests include database, distributed databases, blockchain, and formal methods.

DIVAKAR YADAV received the B.Tech. degree from the G.B. Pant University of Agriculture & Technology, Pantnagar, India, the M.Tech. degree from IIT Kharagpur, India, and the Ph.D. degree from the University of Southampton, U.K., all in computer science. He is a Professor with the Department of Computer Science and Engineering, IET, Lucknow. His research interests includes database systems, distributed computing, formal methods, verification of critical properties of business critical systems, and reasoning about distributed database systems. He was a recipient the Young Scientist Award of the Government of Uttar Pradesh, in 2003, the Distinguished Author Award at India Education Conference, New Delhi, in 2008, and the Commonwealth Scholarship by the Government of United Kingdom, in 2004.

GIRISH CHANDRA received the B.E. degree in computer engineering from the M.M.M. Engineering College, Gorakhpur, India, in 1992, the M.Tech. degree in computer science and engineering from IIT Kanpur, and the Ph.D. degree in computer science from A.P.J. Abdul Kalam Technical University, Uttar Pradesh, Lucknow. He is a Professor with the Department of Computer Science and Engineering, Institute of Engineering and Technology, Lucknow. His research interests include cryptography, distributed computing, databases, and formal methods.