iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VII **Month of publication:** July 2022

DOI: https://doi.org/10.22214/ijraset.2022.45713

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

The Use of Formal Approach and Techniques Applied to Understand and Improve the Blockchain

Utkarsh Sharma
B.Tech in CSE at MNNIT Allahabad

Abstract: ***With data volumes expected to expand exponentially in the next years, protecting that data is critical. To ensure the reliability of the system, we use cutting-edge tools like blockchain. It wasn't until the advent of bitcoin, however, that a sizable portion of the population took notice of blockchain technology. Since its inception, blockchain technology has been used in the academic community and by businesses. What this means is that the information is being stored in a way that makes tampering and unauthorized access very difficult. Blocks, an abbreviation for "blockchain," a digital public ledger, are used to record monetary transactions. All nodes will eventually agree on the same sequence of block attachments, and this is achieved by a process termed "consensus." Learning about the inner workings of each blockchain and the rationale for its unique style of operation is possible via a thorough examination of these algorithms. In this study, I examined and contrasted the different consensus mechanisms now in use with consensus protocols, analyzing their relative relevance in blockchain development. The consensus algorithm behind any specific blockchain-based system is crucial to the reliability and safety of that system. Formal approaches may be used to both increase users' trust in blockchain-based systems and design thoroughly tested and trustworthy solutions. To achieve this level of confidence, consensus methods must be effective. Formal modeling is developing a system in a mathematical language, then thoroughly testing and inspecting it to ensure its correctness. This study examined consensus processes and the role formal approaches play in this space to help with the construction of a trustworthy blockchain-based system. To ensure the accuracy of such procedures, this article discusses the current state of applying formal methods to the blockchain consensus process.***

Keywords: Blockchain, Digital public ledger, Formal verification, Proof-of-Work, Security, computer code

I. INTRODUCTION

Early proponents of blockchain technology, Stuart Haber and W. Scott Stornetta, came up with the idea back in 1991[1], contrarily, Bitcoin's introduction in 2009 [2] The media is profoundly affected. Using cryptographic hashes and a decentralized network of computers, blockchains are a distributed digital ledger that is always up to date. There is no need for an outside party to keep an eye on this. Since it operates without a central authority, it is often referred to as distributed ledger technology (DLT), but the two names are commonly used interchangeably. There are many applications for blockchain technology, and the cryptocurrency industry is only one of them. In addition, the use cases for blockchain continue to expand[3]. Data of all kinds, from social to transactional to sensor, is growing exponentially. International Data Corporation has published a report on the subject (IDC) [4], since this information is being produced at an exponential rate in the year 2020, keeping it secure is one of the most pressing problems we face. The central server concept as it has traditionally been conceived of is insufficient for securing the time-sensitive and safety-critical systems of today. For this reason, a decentralized system is critically needed now so that cynical individuals may interact with one another without having to depend on a centrally trusted third party. Blockchain, a novel and rapidly expanding technology, can meet all of these demands. Each node in a blockchain network verifies information before it is recorded in the blockchain. All nodes must come to a consensus on a means by which new things are added to the chain. It is common to practice to use the blockchain consensus process to verify the veracity and trustworthiness of the recorded data. Another beneficial impact is that the nodes that contributed to adding the new block to the blockchain become more coherent. Blockchain technology resolves the issue of switching from a single, centralized ledger to a network of independent nodes that can be trusted to record and verify transactions transparently and safely. A blockchain can't function without a consensus mechanism. This method specifies the process through which all nodes agree to generate a new block. The consensus process might be seen as important to a blockchain-based system because of the trust it fosters and the resilience it provides against failure.

We explore the many consensus protocols utilized in blockchain technology, the underlying mechanisms that allow it to function, and the potential drawbacks of each protocol, such as the amount of processing power and convergence time necessary. Important features of the underlying consensus algorithms, such as security and resistance to assaults, are also highlighted. These benefits increase the potential application areas for the technology. Byzantine fault-tolerant consensus methods have recently come back into the limelight, thanks to blockchain systems. Given the importance of consensus protocols to the overall functioning of the blockchain infrastructure, ensuring that they function as intended is of paramount importance. Formal processes used in this kind of emerging technology may help ensure the reliable functioning of the system. Using formal procedures early in the creative process is more effective for increasing the impacts of new technologies in the area of information technology. Formal approaches have long been considered the best way to define and document precise system requirements. There is a dearth of literature on applying formal techniques to blockchain technology and the protocols that underpin it since the field is still in its infancy. Blockchain networks are made as secure as possible via the use of formal approaches. Many protocols, not only those involving cryptography, consensus, or security, might stand to gain from the use of such techniques [5].

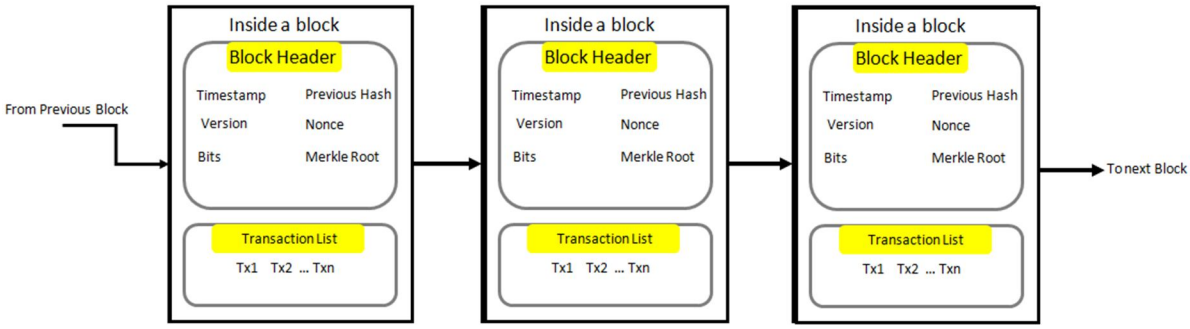


Figure 1: Structure of Blockchain Blocks.

II. THE BASICS OF BLOCKCHAIN TECHNOLOGY AND SMART CONTRACTS

Blockchain is a distributed ledger that stores a complete and immutable record of all transactions in a network and is available to all nodes. To ensure the security of data, blockchains generate cryptographic hashes. Insertion of the blocks occurs in order via atomic broadcast. A consensus mechanism is engaged on all participating nodes whenever a new transaction is processed or a new block is generated, guaranteeing the integrity of the global ledger at all times. This helps ensure that numbers from all across the globe add up correctly. To ensure that no changes have been made to the data, a cryptographic hash is calculated and compared to a known good fingerprint. A block is a compilation of transactions and their associated hashes. To identify a block, a hash of its linked transactions will be created. Each block in the chain is linked to the next by including its hash in the header of the next block. As a byproduct, blockchains are formed. Security in blockchains comes from the chaining together of blocks, which is done using cryptography. These blobs include all of the information. Figure 1 is a graphical representation of the block formation process in a bitcoin blockchain. Both a header and a transaction log keep track of the many exchanges that took place inside a single block. The document's header has the following six sections. (i) Timestamp of when the block was first created. Future blocks on the blockchain may be connected by storing (ii) the hash of prior blocks. The Merkle Root, often known as the transaction root, is the hash sum of all verified transactions. The hash value of each legitimate transaction is calculated, and then those values are added together in pairs with the hash values of the other transactions to generate a new hash. Repeating this method numerous times yields a hash that contains all of the monetary transactions. A Merkle tree is then used to complete the whole process. The version of the protocol utilized by the node recommending the new block (iv). To create a nonce in the Proof-of-Work (PoW) consensus procedure, (v) a difficult mathematical problem must first be solved. (vi) One possible unit of measurement for the difficulty of a Proof of Work issue is the number of "bits," where each bit corresponds to one digit of a decimal number. Harber and Scott were the first to openly propose the idea of a blockchain in 1991 [6], In other words, you are suggesting we discover a method to guarantee the security of papers such that they cannot be altered in any manner. The use of cryptographically secure chains for archiving paper time stamps was suggested. Since its introduction in 1992, Merkle trees have made it possible to store much more documents in the same amount of physical storage. However, no one used the technology until the introduction of Bitcoin in 2008 [7].

Accelerate the uptake of blockchain-based services. Hashcash, the Proof of Work (PoW) algorithm or PoW mechanism, is the second most widely used Bitcoin technology after the blockchain.

By documenting and confirming transactions, this decentralized P2P system safeguards users against accidental double-spending. In 2013, after some time had elapsed, Vitalik Burterin started developing a language for bitcoin development.

He oversaw the development of Ethereum, a distributed computing platform based on blockchain technology and including its programming language [8]. Smart contracts, or blockchain contracts, are a kind of computer code used in digital transactions. A smart contract is an agreement between two parties that specifies the conditions under which a multi-step automated procedure will begin. Nick Szabo first suggested the concept of a smart contract in 1994. A smart contract is just a script stored on a blockchain. To trigger a smart contract, a transaction must be made to the contract's address on the blockchain. Once the transaction details that activate the contract have been transmitted, the contract will be executed automatically and without further human interaction at every node in the network [9]. Because of this, it is reasonable to assume that each node in a smart contract is a virtual computer (VM). To do computations inside the Ethereum network, users may make use of the Ethereum Virtual Machine (EVM). Transactions may be seen and audited in a way that is both visible and auditable with the help of smart contracts. The following may be said about the history of blockchain technology below in Fig.2.

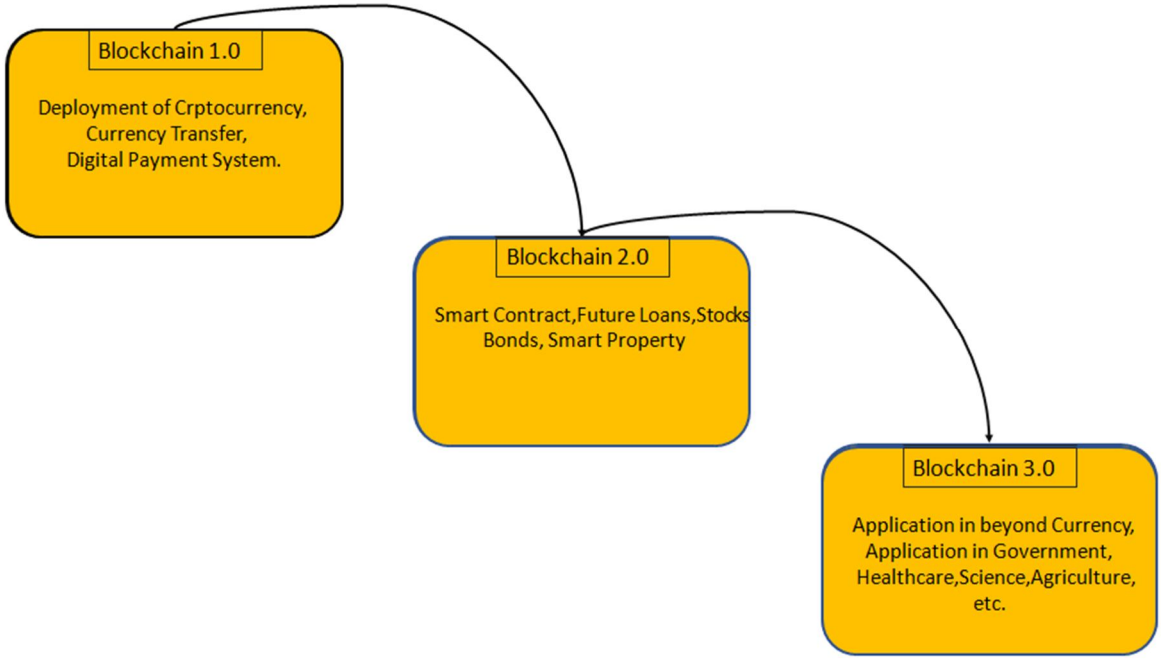


FIGURE 2: Growth of Blockchain Technology.

For reasons outside the realm of cryptocurrencies, blockchain technology is attracting increasing attention. Formalization and verification of blockchain's safety and security elements are becoming more important as it finds widespread use in high-assurance-requirement industries. This is because blockchain technology may be used to record transactions in a distributed ledger. The blockchain network is an example of a secure, distributed system for transferring information. As a result, formal procedures may enhance both user safety and the system's overall dependability [10].

III. EXPLAINING THE WORKINGS OF A BLOCKCHAIN SYSTEM

The Hash Chain, the Merkle tree, and the Digital Signature are the three underpinning technologies essential to the blockchain's operation [11]. To put it another way, these are the three mainstays of the blockchain. (i) kind of storage location indication is the hash pointer. The hash value serves as a checksum that may be used to detect any changes to the original file. The user must alter the hash pointer of each previous block to make modifications to data that has already been stored. (ii)You may be certain that all hash references will be connected when you use a Merkle Tree, also known as a binary search tree. Merkle tree can protect against and foil attempts to corrupt the data. (iii) A digital signature uses cryptography to authenticate the authenticity of data. This method may also be used to ensure the accuracy of the data. The digital signature must be tamper-proof and independently verified. In a decentralized blockchain system, all nodes must agree before a new block can be added to the global chain. The network is updated whenever a new block is created.

Any network node that gets a newly generated block has the option of adding it to its copy of the global ledger. When a critical mass of a network's nodes agrees on a single state transition, the agreement has been achieved. Due to the increasing significance of the blockchain or global ledger, this agreement is crucial.

Several elements, including consensus methods, digital signatures, Merkle trees, and hash chains, may be used to build trust in a blockchain-enabled system. To add a new transaction to the blockchain, a node must first create it and hold it in a pool. It is necessary to get consensus from the great majority of nodes before any new information can be added to the blockchain. The remaining nodes will validate the transaction and add it to a new block after it has been broadcast to the network. Figure 3 is a flowchart of this procedure, which shows how this block will be sent through the network. To ensure the block is legitimate, the remaining nodes employ a consensus-based method. Since it is the consensus process's job to authenticate and validate transactions, it is a crucial element of developing a blockchain. The order of blocks on a blockchain is determined by the consensus methods [12].

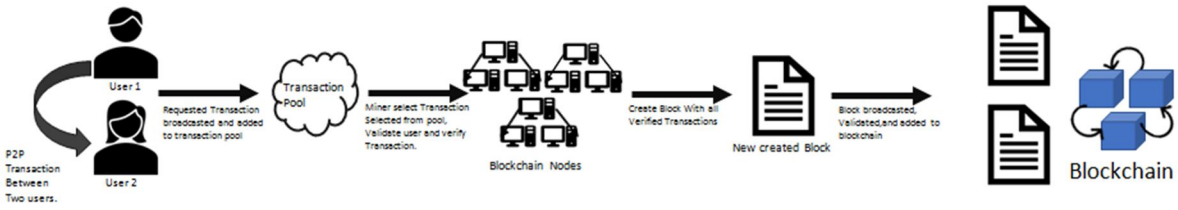


FIGURE 3: Working of Blockchain.

IV. BLOCKCHAIN TYPES

Each blockchain is made up of nodes that engage in decentralized communication with one another. The distributed ledger is updated throughout the network in this way. Nodes can do more than just build blocks; they may verify transactions, relay communications, and function as intermediates. Blocks may also be used to verify financial transactions, send and receive messages, and create new network nodes. Most blockchains may be categorized according to the following criteria for node participation in consensus ledgers:

A. Public Blockchain

There is no dominant entity inside the network. One is at no disadvantage for either remaining in or leaving the network. Any user on the network may check the validity of transactions by accessing the blockchain. A public blockchain system is used by coins like Bitcoin. Miners are responsible for verifying the authenticity of Bitcoin transactions. Bitcoins are given to miners in the form of transaction fees and newly produced Bitcoins for their efforts in solving the mathematical mystery employed in the PoW consensus process. Bitcoins are created and distributed to all users as a kind of incentive for taking part in the system.

B. Blockchain Consortium

In a blockchain managed by a group of people, not all of the nodes are created equal when it comes to their ability to verify trades. Certain nodes in the network are specifically designated to verify monetary exchanges. It's conceivable the rest of them will come around, but keeping the implementation continuing requires at least this proportion of nodes to agree.

C. Private Blockchain

There may be significant differences between the consortium blockchain and the private blockchain. There is a lot of hierarchy in the organization. A central authority makes all the calls and oversees its verification. The centralized head is responsible for ensuring that the final deal is in line with the initial vision. The public blockchain system is referred to as a "permissionless blockchain," whereas the other two kinds of blockchains are referred to as "permissioned blockchains." The advantages of a private blockchain, sometimes called a permissioned blockchain, over a public blockchain, also called a permissionless blockchain, are abundantly evident [13].

D. Blockchain-Based Formal Techniques

Formal methods for modeling, such as process algebra, state-transition modeling, and set-based approaches, may be used, and the accuracy of these models can be verified, via the use of formal verification tools.

There is always the option of using a formal method to validate any claim. In this section, we'll discuss the formal techniques that are important to the operation of blockchains.

V. THE STEPS OF ALGEBRA

The steps of algebra [14], model the behavior of distributed or parallel systems as a set of interacting concurrent processes using a wide range of mathematical methods. Developing a model of such a system is essential for gaining insight into its inner workings. Process algebra is one of several mathematical methods that may be used to completely understand the meaning of a syntactically correct process. There are other options available. It contains a set of constructors, equational axioms, and operational semantics, and it explains system development in terms of labeled transitions. Furthermore, assumptions from the field of operational semantics are included [15]. Here, we'll examine the process algebra that might one-day power blockchain networks.

A. Pi-Calculus

Pi-calculus, or calculus as it is more often known, is a concise language with rich expressiveness despite its restricted vocabulary. It enables the encoding of functioning programs. Since the PI-calculus has proven beneficial in studying cryptographic protocols, it is now being used in the formal study of consensus processes in blockchain technology [16]. By going through this procedure, you will have access to the natural calculus, which may be used in your consensus protocol. The suggested calculus is used to write out the specifications of consensus protocols [17].

1) *Communicating Sequential Process(CSP):* CSP [18], In this language, mathematical and logical ideas are used to analyze the interplay of many systems. The notion of concurrent systems programming revolves around the exchange of messages between running processes and systems (CSP). The logic and mathematics behind this information exchange are solid. In 1978, Tony Hoare pioneered the notion of CSP. Using CSP, it is possible to analyze computer systems, programs, and even language. Like process algebra, constraint satisfaction problems (CSP) use algebraic equations and logical reasoning to analyze and define connections between processes. As a subset of constraint fulfillment issues, process algebra is an important area of study in logic (CSP). It might be used to make sure that several threads in a blockchain-enabled program are in sync with one another [19].

2) *BitML:* This is a high-level language for smart contracts since it provides a computationally sound Bitcoin embedding and a thorough and sound technique for checking essential trace features. BitML paves the way for the development of several smart contracts, which may be implemented by having the appropriate transactions recorded on the Bitcoin blockchain. R. Z. Roberto and M. Bartoletti. Created a set of tools for creating, validating, and enforcing BitML contracts on the Bitcoin network. Both security analysis and the validation of arbitrary LTL properties stand to benefit from the suggested method [20].

B. State Transition

To study computers from a theoretical perspective, researchers might use a notion known as a transition system. It's a means to describe the behavior of various systems under isolated conditions. Each state and transition may be labeled, with the same label possibly being used for many transitions. In this part, we will examine the many state-transition modeling techniques that may be used for system analysis.

1) *Petri nets:* Petri nets are a kind of bipartite network used to model and explain complex processes. There are two kinds of nodes in this structure: endpoints and connecting pieces. For those who prefer a somewhat different moniker, "P/Tnets" stands for "place transition nets" and is another term for Petri Nets. Nodes in a network may be connected by the use of directed arcs. In transition, pre-arcs act as entrances and post-arcs as exits. In all cases, the arcs represent transitions between states. The fact that a Petri Net may be described in algebraic formalism is one of its main advantages. That's one of the advantages. Two areas of inquiry into which the blockchain technology was used were addresses and financial transactions. To simplify the creation of an algebraic Petri Net representation, we provided an explanation of each of these components in terms of set theory [21].
2) *Time-Automata(TA:* A finite automaton model with clock variables is shown. Formal modeling may be accomplished with the help of timed automata, which are built from the ground up on the composition of the simple clock and state restrictions (TA). Several model-checking methods are based on timed automata, which are used to evaluate real-time systems. The study presents a modeling framework for Bitcoin contracts that are based on timed automata. The proposed model is then validated with the help of the Uppaal model verification tool [22].

3) *Markov Decision Processes(MDP):* A Markov Decision Process, also known as an MDP, is essentially a mathematical framework that is used for decision-making in transition models, where the Markov model is employed to define the transition models. The MDP may be used in the process of analyzing how well blockchain systems function [23], In this article, we will examine the process of adding new blocks to the blockchain. It has been said that a user must wait for his block to be added to the chain before proceeding. In this work, we use the Markov queue model to improve the transaction confirmation rate and shorten the time between blocks. Some mathematical modeling studies have utilized MDP as a starting point to determine the best selfish mining strategy for blockchain networks [24], [25]

C. *Set-Based*

Functional components of administration, such as Event-B [26] and TLAC [27] Models with their roots in set theory and logic are used to represent programs at a higher level, which is often referred to as modeling the system's executional element. That research analyzed the steps that went into making the system [28Please provide official verification of the Event-B smart contract's legitimacy. The point is to make sure the smart contracts everyone uses are safe, correct, and carry out their intended functions as programmed [29], Once the smart contract had been translated into Event-B models, it was validated on the RODIN platform. The TLAC framework may be used to verify the integrity of the cryptographic protocols and consensus mechanisms utilized by a blockchain [30]. In addition to the above-mentioned techniques, the following may be used to formally write or model the blockchain-enabled system. Diagrams of Data Flow, Control, and Input/Output I: By following the graphs' paths, we can see how the program is expected to run in its entirety. One of the most essential aspects of every piece of software is its building blocks or basic blocks. (i) A basic block is a collection of operations that, barring an exception inside the block itself, may safely be executed in parallel. If the operation generates an exception, the basic block will not execute. In many graphical representations, branching in the control flow is represented by a directed edge. (ii). The Abstract Syntax Tree (AST) notation provides a tree-like representation of a blockchain system. Blockchains may be seen as a hierarchy of abstract data types, each with its own set of consistency constraints that determine the eventual state of the system as a whole. (iii). Linear temporal logic, to give it its full title, is often used in the system design process at the specification stage (LTL). Each of its propositional variables, logical operators, and temporal model operators is selected with great care. The number of propositional variables is restricted to a limited set. Trust between the leader and the validator is crucial to the success of any consensus process. Linear temporal logic (LTL) is the official language for expressing all of these, and it provides security against a wide range of potential threats. (iv) Branching-time logic is used in several places throughout CTL (Computation Tree Logic). A potentially chaotic tree-like structure. Any number of potential outcomes lie in wait for the future. It has several applications, including the formal verification of software and hardware systems. Model checkers are used to verifying the existence of safety and liveness properties in software systems. Users may use CTL to signal that all further program executions should be paused to prevent a potentially disastrous outcome (such as the division of an integer by zero) (e.g., dividing a number by zero). By examining all possible transitions between program states that meet the requirements for the starting condition, a model checker can ensure that the resulting executions all adhere to the safety property. Computation tree logic, a branch of time-based logic, is linked to LTL (LTL). A system may be modeled or formally described via the process of creating a written specification, which can then be verified. The efficiency with which a model is verified is most affected by the methodology or formal model approach used to construct the specification or model the system. We use symbolic notation and program verification to assess modeled systems, in addition to the more well-known Model Checking and Theorem Proving techniques. To guarantee precision after encoding temporal attributes in TLAC, a model checker is used. Systems designed in CFG are tested using symbolic execution techniques to guarantee their correctness. Hoare logic is often used in the formalization of theorem proving. It is used rather often, in contrast to other methods of verification. For this reason, model-checking is a part of the verification process [31], [32].

D. *A Formal Approach for Consensus Protocols*

In the preceding part, we discussed the many consensus protocol variants, such as voting-based and proof-based consensus methods. When compared to proof-based consensus protocols, we observed that voting-based consensus protocols, which are employed in newly developed private and consortium blockchains, are superior. The voting-based system provides greater error tolerance, lower processing requirements, and better throughput. This increases the odds of success for everyone. Voting-based techniques improve the algorithmic and practical feasibility of reaching a consensus. As a result of their many advantages, formal approaches are often given a lot of weight in vote-based consensus processes. Due to the immaturity of the field, we were only able to locate a handful of publications dealing with the formal modeling and verification of consensus procedures.

Our first efforts were directed at absorbing data about the PAXOS [33] Lamport thought of and developed the first algorithm specifically designed for reaching consensus in an asynchronous system. In addition, it provides a foundation for blockchain's consensus mechanisms. This algorithm is very effective and lenient. When compared to this, however, it first failed to persuade other scientists. Time passed, and Lamport revamped the PAXOS OS, which is now extensively used by platforms like the Google _le System [34]. A formal presentation is presented so that you may learn everything there is to know about PAXOS in [35], This article presents a thorough analysis of the theoretical foundations of time automata, focusing on the Clock General Timed Automaton (Clock GTA) architecture. It does it by logically and methodically laying out each stage of the procedure. Following this, in [36], The PAXOS specification is written in the Promela programming language, and the formal modeling of PAXOS is given in the form of finite state automata. It is a high-level language with precise, machine-executable semantics that allows you to express guards consistently. We also utilized the SPIN model checker to double-check the model's precision. Raft, a kind of voting popular in the business sector, is also used. Raft is sometimes seen as an extension of PAXOS, another popular private blockchain platform.

It's more versatile and easier to use than PAXOS, for starters. It's a part of Zookeeper, much like Google and Facebook. The Raft consensus approach has opened up a wealth of possibilities in formal modeling. Since Raft was published, the TLAC has made public an imperfect formal specification of the Raft protocol[37]. In [38], The previously released formal definition of Raft has been revised to incorporate new features that serve as a safety net throughout the algorithm's leader election procedure. Because of these efforts, Raft protocol dictates that only one leader may hold the position of captain at any one moment. State machine safety is the most crucial safety characteristic, and this paper presents the first officially confirmed implementation of this safety feature using the Raft approach [39]. It has been formally verified that the protocol is linearizable and that each replicated state machine contains the same sequence of events and executes the same instructions in the same order. In [40], It is shown that Raft may be formally modeled in LNT process algebra, and this is verified using computer-aided design proofs [41], Model-checking strategies. The CADP software may be used to visualize the specified transition system. The author of the official TLAu definition of Raft noticed a state transition from candidate to follower in this particular section. In [42], For RAFT validation in TLA, we use the IPA architecture. It has been observed that the Raft protocol's log replication and leader election occur in two independent periods. If the elected leader is incapacitated, the cluster may remain unreachable until a replacement is chosen or the incumbent is re-elected. In [43], The TLAu modeling language was used to create formal models of this situation, which were then validated using the TLC model checker.

Successfully testing complicated systems by verifying models has been shown. Its usefulness as a consensus mechanism in asynchronous distributed systems is constrained by the fact that different consensus protocols might be in several different states. In addition, it is impractical to do tests in every possible situation. These consensus techniques need to be rigorously validated, and this can only be done with the help of formal verification that is straightforward to implement. In [44], The consensus method of an asynchronous system may be extensively examined through a computational model based on the Heard-Of concept (HO model). The paradigm allows for a high level of abstraction. There is no work being done to formalize BFT verification. in [45] By using ByMC, a model checker. Steller, a quorum-based consensus algorithm, was inspired by the BFT consensus method. The Stellar protocol's security and viability may be verified using the methods provided [46]. The protocol is written in first-order logic and validated using a combination of the Isabelle/HOL and Ivy proof assistants. To ensure their efficacy, consensus-building processes must be explicitly codified and rigorously tested. One of the most used formal verification techniques, model checking, is based on formal processes. If a formal model of the system exists, it may be verified for conformance to the requirements. Model checking is better than conventional testing and simulation because of its ability to reveal deep-seated errors [47].

VI. CONCLUSION

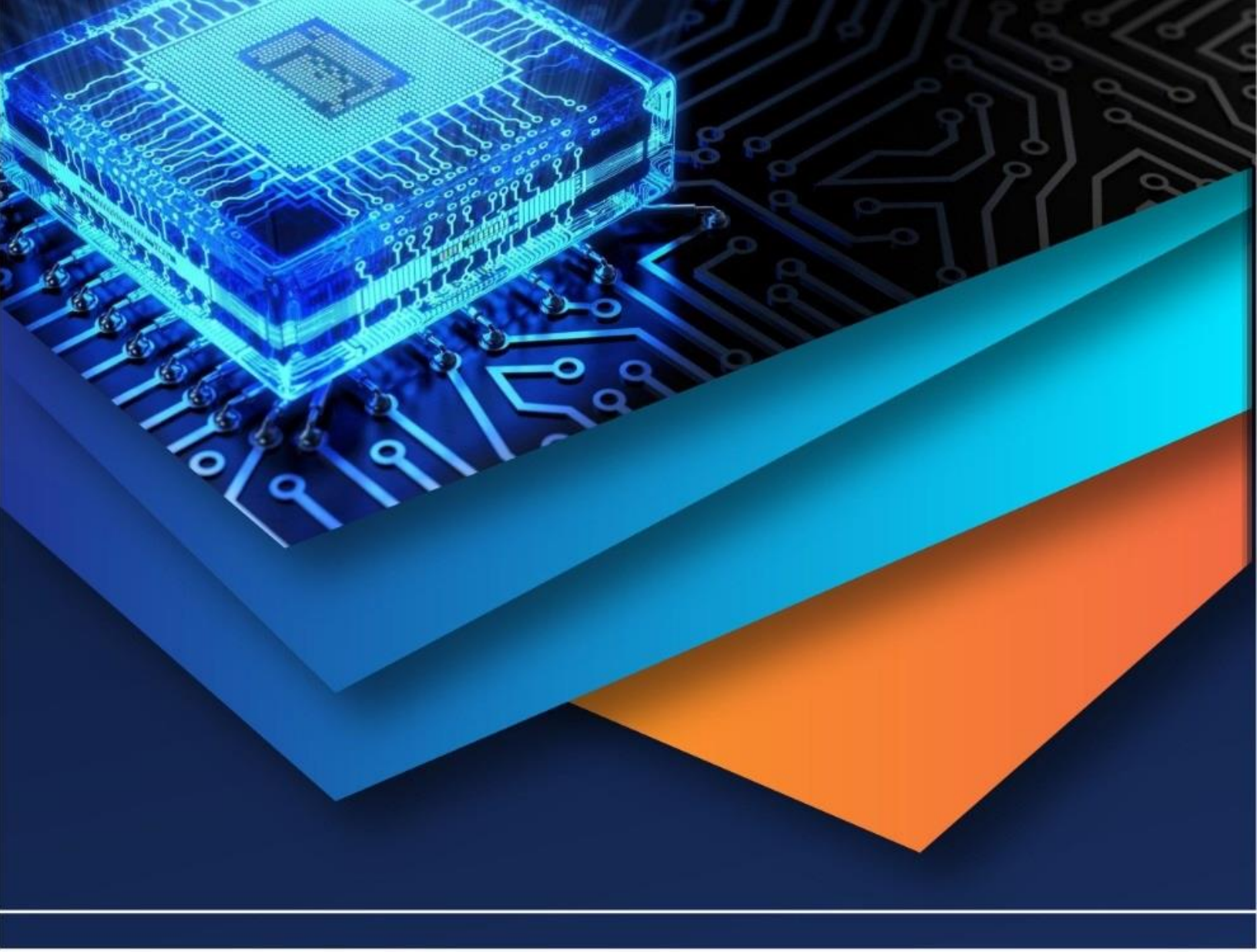
The blockchain system is now one of the most discussed technologies. Since blockchain technology and smart contracts are still in their infancy, there is no commonly agreed standard or best practice for officially confirming them. Businesses operating on blockchains benefit from increased trust and transparency when their consensus algorithms have been formally verified. In this article, we look at some of the potential formal approaches that might be used in conjunction with blockchain-enabled systems. To establish confidence in a network, we have suggested that formal specifications for consensus protocols and blockchain-enabled systems are required.

After much investigation, we know that model checking is the most popular method for ensuring the safety of blockchain-based systems. In the not-too-distant future, the incorporation of formal techniques into this widely used technology will become increasingly essential and open up new lines of investigation.

REFERENCES

[1] S. Haber and. 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, Framingham, 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] S. Haber and. S. Stornetta, ``How to time-stamp a digital document," in Proc. Conf. Theory Appl. Cryptogr. Berlin, Germany: Springer, Aug. 1990, pp. 437_455
[7] S. Nakamoto and A. Bitcoin. A Peer-to-Peer Electronic Cash System. Accessed: Mar. 4, 2022. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[8] V. Buterin, ``A next-generation smart contract and decentralized application platform," White Paper 3, 2014, no. 37.
[9] 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.
[10] Sharma, Utkarsh. "blockchain technology- a conceptual overview," International Research Journal of Engineering and Technology (IRJET), March. 2022, e-ISSN: 2395-0056, p-ISSN: 2395-0072. https://www.irjet.net/archives/V9/i3/IRJET-V9I3321.pdf
[11] N. Szabo, ``Smart contracts," Virtual School, Tech. Rep., 1994. Accessed: Mar. 17, 2022. [Online]. Available: http://szabo.best.vwh.net/ smart.contracts.html
[12] R. Beck, ``Beyond bitcoin: The rise of the blockchain world," Computer, vol. 51, no. 2, pp. 54_58, Feb. 2018.
[13] I. Bashir, Mastering Blockchain. Birmingham, U.K.: Packt, 2017.
[14] 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.
[15] 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.
[16] W. Fokkink, Introduction to Process Algebra. Cham, Switzerland: Springer, Dec. 1999.
[17] 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.
[18] P. Tolmach, ``A survey of smart contract formal speci_cation and veri_cation," ACM Comput. Surv., vol. 54, no. 7, pp. 1_38, 2021.
[19] 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
[20] 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. Secure., vol. 102, Mar. 2021, Art. no. 102098.
[21] M. Bartoletti and R. Zunino, ``BitML: A calculus for Bitcoin smart contracts," in Proc. ACM SIGSAC Conf. Comput. Commun. Secure., Oct. 2018, pp. 83_100.
[22] A. Pinna, ``Petri nets model for blockchain analysis," Comput. J., vol. 61, no. 9, pp. 1374_1388, 2018.
[23] R. Srivastava, ``Mathematical assessment of blocks acceptance in the blockchain using Markov model," Int. J. Blockchains Cryptocurrencies, vol. 1, no. 1, pp. 42_53, 2019.
[24] J. Niu and C. Feng, ``Sel_sh mining in Ethereum," 2019, arXiv:1901.04620.
[25] C. Grunspan and R. Pérez-Marco, ``On pro_tability of sel_sh mining," 2018, arXiv:1805.08281.
[26] J. R. Abrial, Modeling in Event-B: System and Software Engineering. Cambridge, U.K.: Cambridge Univ. Press, May 2010.
[27] L. Lamport, Specifying Systems: The TLAC 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
[28] J. Zhu, K. Hu, M. Filali, J.-P. Bodeveix, and J.-P. Talpin, ``Formal veri_- cation of solidity contracts in event-B," 2020, arXiv:2005.01261.
[29] A. Lahbib, ``An Event-B based approach for formal modeling and very _cation of smart contracts," in Advanced Information Networking and Applications. Cham, Switzerland: Springer, 2020.
[30] Sharma, Utkarsh. "Internet-of-things (IoTs) architecture and its diverse layers affect safety, transparency, and integrity." International journal for research in applied science and engineering technology 10.4 (2022): 187-200.
[31] V. Kukharenko, ``Veri_cation of HotStuff BFT consensus protocol with TLAC/TLC in an industrial setting," in Proc. Comput. Sci. On-Line Conf. Cham, Switzerland: Springer, 2021, pp. 77_95.
[32] M. Almakhour, L. Sliman, A. E. Samhat, and A. Mellouk, ``Veri_cation of smart contracts: A survey," Pervez. Mobile Comput., vol. 67, Sep. 2020, Art. no. 101227. VOLUME
[33] Z. Neha, 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.
[34] L. Lamport, ``Paxos made simple," ACM SIGACT News Distrib. Comput. Column, vol. 32, no. 4, pp. 51_58, Dec. 2001.
[35] M. Burrows, ``The chubby lock service for loosely-coupled distributed systems," in Proc. 7th Symp. Operating Syst. Design Implement., 2006, pp. 335_350.
[36] R. De Prisco, B. Lampson, and N. Lynch, ``Revisiting the Paxos algorithm," Theor. Comput. Sci., vol. 243, nos. 1_2, pp. 35_91, 2000.
[37] G. Delzanno, M. Tatarek, and R. Traverso, ``Model checking Paxos in spin," 2014, arXiv:1408.5962.
[38] D. Ongaro, Consensus: Bridging Theory and Practice. Stanford, CA, USA: Stanford Univ., 2014.
[39] 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
[40] D.Woos, J. R.Wilcox, S. Anton, Z. Tatlock, M. D. Ernst, and T. Anderson, ``Planning for change in a formal veri_cation of the Raft consensus protocol," in Proc. 5th ACM SIGPLAN Conf. Certi_ed Programs Proofs, Jan. 2016, pp. 154_165.
[41] H. Evrard, ``Modeling the raft distributed consensus protocol in LNT," 2020, arXiv:2004.13284.

[42] 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.
[43] X. Gu, W. Cao, Y. Zhu, X. Song, Y. Huang, and X. Ma, ``Compositional model checking of consensus protocols speci_ed in TLAC via interaction preserving abstraction," 2022, arXiv:2202.11385.
[44] G. Yu, L. Hua, L. Yuanping, L. Bowie, W. Xianrong, and R. Hongwei, ``Using TLAC to specify leader election of raft algorithm with consideration of leadership transfer in multiple controllers," in Proc. IEEE19th Int. Conf. Softw. Qual., Rel. Secure. Companion (QRS-C), Jul. 2019, pp. 219_226.
[45] B. Charron-Bost and S. Merz, ``Formal veri_cation of a consensus algorithmin the heard-of model," Int. J. Softw. Information., vol. 3, nos. 2_3, pp. 273_303, 2009.
[46] P. Tholoniat and V. Gramoli, ``Formal veri_cation of blockchain byzantine fault tolerance," 2019, arXiv:1909.07453.
[47] G. Losa and M. Dodds, ``On the formal veri_cation of the Stellar consensus protocol," in Proc. 2nd Workshop Formal Methods Blockchains (FMBC), 2020, pp. 9:1_9:9.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY