1

Introduction

Blockchain has emerged as a prominent area of research amongst researchers and academicians after the huge success of Bitcoin cryptocurrency [1]. The distinguished feature of blockchain is that it enables a fully decentralized system, meaning no single entity (node) governs or controls the blockchain operations [2]. Apart from decentralization, blockchain also offers an immutable, transparent, and reliable distributed ledger on the ordered set of transactions by leveraging cryptography and carefully designed incentive schemes [3]. Blockchain achieved all these features, i.e., decentralization, immutability, transparency, and distributed ledger using the concept of consensus algorithms [4]. As blockchain technology matured, researchers sensed its applicability far beyond digital currencies, leading to programmable blockchains like Ethereum [5]. In the programmable blockchain, a novel concept is introduced called a smart contract [6], a self-triggered program code with the terms of the agreement directly written into the code. This capability opened up possibilities for automating complex processes and creating decentralized applications (Dapps) that operate on the blockchain [6]. These developments have significantly broadened blockchain's applicability, positioning it as a foundational technology for various decentralized systems beyond cryptocurrencies [7]. With the capacity to stimulate trustless transactions and its inherent resistance to tampering, blockchain is poised to redefine a host of industries, such as the Internet of Things (IoT), Intelligent Transportation Systems (ITS), and energy distribution in Smart Grid [8]. Its promise lies in creating new, decentralized economic and social infrastructures where direct peer-to-peer interactions are not only possible but also reliable, secure, and efficient. Despite the hype, promising features, and vast applications, blockchain technology still faces many significant challenges, such as inefficient consensus algorithms, limited performance and scalability, absence of interoperability, and concerns about privacy [9]. In this thesis, we aim to address all these challenges. Before a deep dive into the thesis, the next section defines four basic building blocks of blockchain: block, blockchain, node, and Merkle root.

1.1 UNDERLINE BLOCKCHAIN TERMINOLOGY

Blockchain's data structure is ingeniously designed to be both transparent and secure. Each block within the chain contains a header and a body. The header includes the Block Number (B_{ID}), Merkle Root Hash (Mrt), Timestamp (T), Block Hash (BH), and Previous Block Hash (PBH). The body contains lists of individual transactions. Each block is connected to its previous block using the hash cryptographic technique called Secure Hash Algorithm (SHA). The security of the blockchain comes from this cryptographic hash function, which ensures any alteration of transaction data within a block is nearly impossible because it

requires altering all subsequent blocks after that, which is computationally infeasible on a distributed network. Further, public-key cryptography allows users to transact on the blockchain with a pair of keys. A public key serves as an address on the blockchain, and a private key is used to sign transactions, proving ownership of blockchain assets. Typically, blockchain is a linear structure very similar to the linked list, i.e., it stores all the transaction history bunched in the blocks, which are linearly connected. Figure 1.1 shows the blockchain structure.

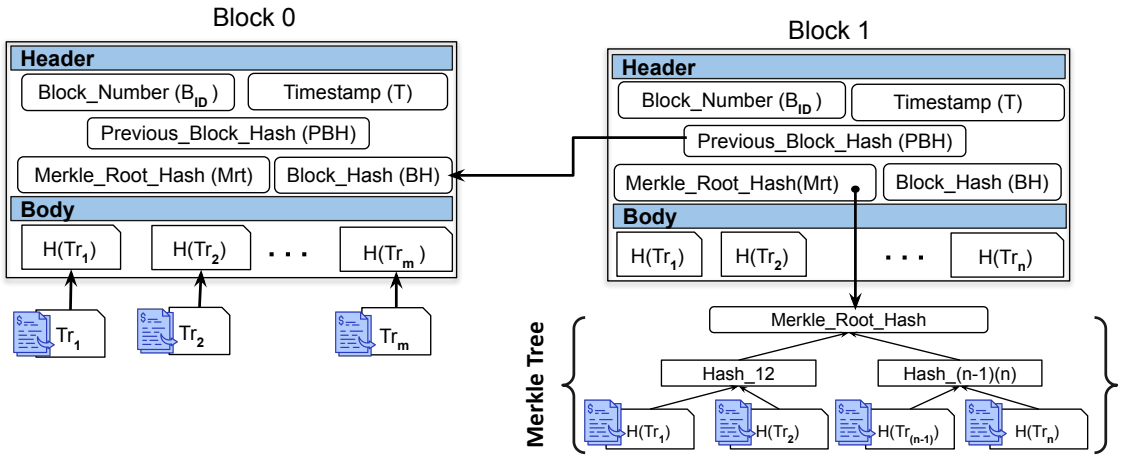


Figure 1.1: Typical graphical representation of blockchain

Definition 1 (Block). *A block is the first and basic component of any blockchain. Here, we denote the block as B. As shown in Figure 1.1, a block is generally a tuple of six elements as* $\sum_B = (B_{ID}, T, BH, PBH, Mrt, \mathbb{S}|_{Tr})$. *The* B_{ID} *is a unique block identifier or number, and* T *is the timestamp. BH and PBH are the block hash and previous block hash, respectively. The next tuple,* Mrt*, is the Merkle root of the set of transactions in the block. The* $\mathbb{S}|_{Tr}$ *is the set of transactions. In cross-chain communication, the transactions include the asset (*$Ast \in \mathbb{S}|_{T}r$*) of users transferred from the source blockchain to the destination blockchain.*

Definition 2 (Blockchain). *A blockchain is the collection of orderly connected blocks in a fixed topological order as* $B_1 \leftarrow B_2 \leftarrow B_3 \leftarrow ... \leftarrow B_N$. *It is a tuple of two elements, i.e.,*$\sum_{Blochain} = (Gns, B_L)$, *where Gns is the genesis block and* $B_L = [B_1, B_2, B_3, ..., B_N]$ *is the list of blocks. The blocks are cryptographically linked in such a way that the hash of* B_1 $(H(B_1))$ *is included in the* B_2*, the hash of* B_2 $(H(B_2))$ *is included in the* B_3*, the hash of* B_{N-1} $(H(B_{N-1}))$ *is included in the* B_N*, as shown in Figure 1.1.*

Definition 3 (Node). *A node is a high-performance processing unit within a distributed network responsible for creating blocks and adding them to the blockchain ledger. These nodes are referred to as the "Miner" nodes. There are two types of nodes in various blockchain technologies, such as Bitcoin. One is a full node, which maintains a complete copy of the blockchain, while the other is a partial node, which relies on the full node to commit transactions or assets.*

Definition 4 (Merkle Root). *The Merkle tree, named after its creator, Ralph Merkle, is a fundamental component of blockchain technology [10]. It encapsulates the summary of all transactions in the block as shown in Figure 1.1. It is a type of binary tree composed of hash-based nodes, i.e., leaf nodes, intermediate nodes, and single root nodes. The root node also referred to as Merkle Root, is denoted as* Mrt*. For a block B containing a set of transactions* $\mathbb{S}|_{Tr} = Tr_1, Tr_2, ..., Tr_n$*, the* Mrt *is computed, where each leaf node represents the hash of a transaction* $H(Tr)$*, each non-leaf node represents the hash of the concatenation of*

its two child nodes, denoted as $H(C_{left}||C_{right})$. This computation proceeds recursively from the leaf nodes to the root of the tree, formally defined as $Mrt = H(...H(H(Tr_1)||H(Tr_2))...||H(...H(Tr_{n-1})||H(Tr_n)))$ for an even number of transactions. The last transaction is duplicated for an odd number of transactions to ensure a complete binary tree. The MerkleRoot serves as a cryptographic fingerprint for all transactions in B, enabling efficient and secure verification of transaction data.

1.2 CONSENSUS ALGORITHMS

The main objective of the consensus algorithm is to solve the byzantine general problem. That is, a system must be able to continue functioning correctly even if some of its nodes fail in arbitrary ways. The Byzantine Generals Problem happens when the members of a network do not agree on the same data values. This could be because some members might be sending out wrong messages on purpose or making mistakes because their systems have crashed or are not working properly. Because of these issues, it's really hard for everyone to come to an agreement.

A consensus algorithm is a critical component of blockchain technology, providing true decentralization. In its simplest form, a consensus algorithm is a general agreement on a single data value (e.g., transactions) among distributed processes or blockchain nodes. These algorithms are necessary for ensuring the reliability and integrity of the transactions despite the fact that some nodes behave maliciously. Thus, before appending a new block in the ledger, the majority of the nodes must agree on the correctness of the block. Various consensus algorithms have been developed for different blockchain platforms. The widely known consensus algorithms are Proof-of-Work (PoW) [1], Proof-of-Stake (PoS) [5], Delegated Proof-of-Stake (DPoS) [11], and Practical Byzantine Fault Tolerance (PBFT) [12]. The PoW, PoS, and DPoS are suitable for permissionless networks, while permissioned networks use different categories of consensus algorithms, such as PBFT. The consensus algorithms are broadly divided into two categories, i.e., proof based consensus algorithm and vote based consensus algorithm [13].

1.2.1 Proof Based Consensus Algorithms

Proof based consensus algorithms are the ones where the participating nodes need to generate some proof or prove the commitment of a certain kind of resource to the network to show that it deserves to create the block and append it to the ledger. Often, the permissionless network adopts this category of consensus algorithms. PoW and PoS are two famous consensus algorithms in this category. In PoW, participating nodes have to solve a highly computational mathematical puzzle to show proof of the correctness of the newly created block. When other nodes in the network agree to the generated proof, the block is added to the ledger [1]. In contrast to the PoW, PoS used the concept of putting a stake as high as possible to become a validator. The node with the highest stake is most likely to be selected as the validator to create the block [14]. Below we have defined these two consensus algorithms in detail.

Proof-of-Work

Bitcoin is the first blockchain platform that used the PoW consensus algorithm to verify and process transactions. The consensus algorithm determines which consensus node (also called the miner) appends the newly created block into the ledger to get the rewards for processing the transactions. Figure 1.2 depicts

the working process of PoW. As shown in the figure, there are two users, A and B, who make transactions that are transferred to the corresponding miners. Following the PoW procedure, one of the miner nodes processes these transactions along with another set of transactions, creates the block, and appends them to the ledger. The PoW prevents miners from immediately appending the new bunch of transactions as a new block on the chain. If this immediate appending block is permitted, many miners could simultaneously grow the chain, making it difficult to determine the ordering of blocks. Instead of the immediate appending block, all the miners must solve the cryptographic puzzle to satisfy the threshold value [1]. Each newly appended block must contain a nonce (a random quantity) to find the block's cryptographic hash that is less than the fixed threshold value. The one-way hash function can not be inverted; due to this property, the only way to find a nonce to satisfy the threshold value is through brute-force search. The miners who first solve the puzzle and win the race append the new block to the chain and are awarded some Bitcoins or other associated cryptocurrency. In the PoW, all the blockchain network peers append the newly generated transactions in the transaction pool and set their status as pending (transactions not included in any of the previous blocks). The miners collect some sets of transactions from the transaction pool that have pending status and create a hash tree to calculate the Merkle root. Miners get the previous block's hash, set the timestamp T, and the nonce equals 0. After setting these parameters, it searches for the hash value with different values of the nonce, which satisfy the threshold hash value. If the hash is found before any other miner, it creates the block using all the parameters, including block hash BH, and broadcasts it to the network for verification. All the other miners receive and verify the block; if it is validated, append it to the ledger.

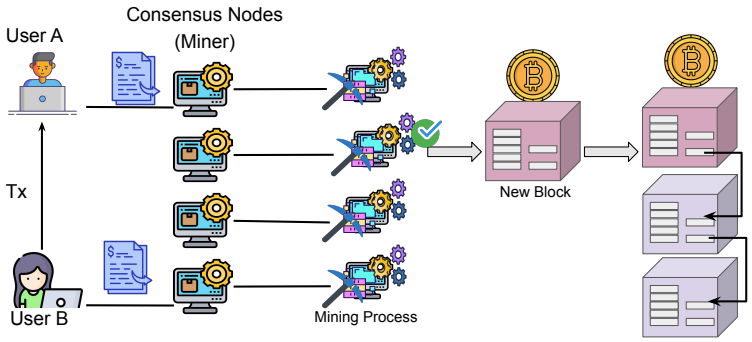


Figure 1.2: Working mechanism of PoW

Proof-of-Stake

A PoS consensus algorithm was proposed to overcome PoW's drawbacks. The PoS also works in the public blockchain, offering protection against attacks. Figure 1.3 shows the typical depiction of the PoS. Similar to PoW, in PoS, there are two users, A and B, who make transactions and transfer to the corresponding consensus nodes. These consensus nodes put some stake in becoming the potential validator for processing transactions and creating new blocks. Based on the highest stake and randomization, one of the nodes is elected as the miner who creates the next block. In PoS, a miner who adds a new block to the blockchain is rewarded as a transaction fee. As opposed to PoW, in PoS, a node becomes a validator by staking cryptocurrency as a security deposit. A new miner is periodically selected as a validator to create and broadcast a new block into the network. All the nodes must submit a security deposit in advance to become a potential validator. The PoS combines the quantity of stakes with one of the two factors to make the selection unbiased and trustworthy for each node on the network. First, coin-age (oldest nodes have the

highest chances to become a new validator), and second, randomization (node which has the lowest hash value and highest stake). In PoS, all the peer nodes put new transactions into the transaction pool. Then, all the nodes who contest to become a validator for the next block raise a stake. This stake is combined with either a coin-age scheme or randomization scheme and node status to select the validator. After getting selected as a validator, the node collects all the pending transactions from the transaction pool, verifies them, creates a block, and broadcasts it to the network for verification. Meanwhile, its stake is locked, and the forging reward (i.e., transaction fees) has not yet been awarded. This step is just a precautionary measure from malicious nodes. If other nodes verify the block, the validator's stake is unlocked, and the reward is awarded. If the coin-age scheme is used for validator selection, then the validator's coin-age is set to 0 for the next round of validator selection so that other nodes also get a chance to become validators, and this makes the scheme fairer to all the nodes. However, if the invalid block is reported as valid, the stake is deducted from the validator and marked as a bad node.

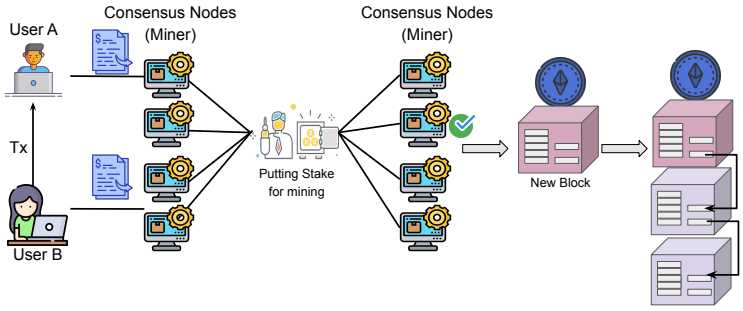


Figure 1.3: Working mechanism of PoS

1.2.2 Vote Based Consensus Algorithm

Unlike proof-based consensus algorithms, Vote based consensus algorithms are the ones that achieve consensus through selection mechanisms, votes, or authority delegation. The selected authority or node creates the block and appends it to the ledger. PBFT and DPoS are the two widely adopted consensus algorithms in this category. PBFT is designed to solve the byzantine general problem. To reach a consensus, PBFT requires a majority (typically $2/3^{rd}$) of the nodes to agree on the state of the currently processing transactions. There are three major stages (i.e., pre-prepare, prepare, and commit) of communication between nodes for the finality of the transactions [12]. In contrast to PBFT, DPoS [11], the stakeholders of the blockchain network elect a number of delegates who are responsible for validating transactions and creating new blocks in the blockchain. DPoS aims to improve the performance by limiting the number of nodes involved in the consensus process. As a result, DPoS has faster transaction processing than PoW and PoS. Below we have defined both these consensus algorithms in detail.

Practical Byzantine Fault Tolerance

Traditional Byzantine Fault Tolerance algorithms were considered to be impractical for real-world applications due to their computational and communication overheads. Castro M. and Liskov B. [12], [15] introduced PBFT as a solution to this problem. Their algorithm was designed to be efficient and practical for real-world systems, ensuring that a distributed service remains available and correct as long as the number of malicious nodes does not exceed one-third of all nodes. PBFT achieves this without relying on

cryptographic assumptions, making it suitable for a wide range of applications. The PBFT protocol operates in a system with a predefined number of nodes, one of which acts as the primary (or leader) node, and the others serve as replica nodes. The primary responsibility of the primary node is to order the requests for service in a sequence and then broadcast them to the replica nodes. The client or user waits for a minimum number of replies from replica nodes (typically a majority) before accepting a response to ensure it is receiving a consistent answer. The protocol operates in a series of rounds, and in each round, the following steps are taken, as shown in Figure 1.4.

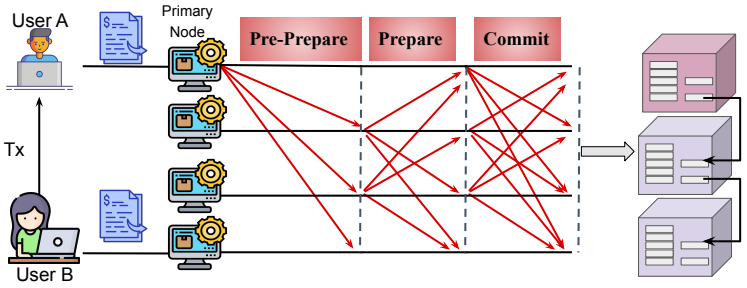


Figure 1.4: Working mechanism of PBFT

1. **Pre-prepare**: As shown in Figure 1.4 this step starts with a leader or primary node, which is periodically changed to ensure fairness and security. This designated leader broadcasts a *pre-prepare* message to the replica nodes in the network, which contains information about the newly proposed block and the current view number. This phase aims to inform the nodes of the new proposal and make a consensus.

2. **Prepare**: Upon receiving the *pre-prepare* message, the replica nodes broadcast a *prepare* message, which indicates that they have received the proposal from the leader node and are ready to execute it. The *prepare* message includes the digest of the proposed block, serving as an endorsement of the proposal's integrity and the leader's legitimacy. To move forward for the next step, i.e., *commit*, there should be positive responses from at least $2/3^{rd}$ of all nodes, i.e., $2f+1$ nodes, where f are the faulty nodes. This step serves as a safeguard against faulty or malicious leaders proposing invalid blocks.

3. **Commit**: Once a node has received 2/3 prepare messages from all nodes (including its own), it broadcasts a *commit* message indicating its readiness to *commit* the proposed block to its local copy of the blockchain. The proposed block is committed to the blockchain if the node receives *commit* messages from $2/3^{rd}$ of all the nodes. This shows that the majority of the nodes agree to the proposed block. The *commit* phase finalizes the consensus process by ensuring that a sufficient number of nodes have agreed to the new block. Once committed, the block is considered final, and the system moves to the next round of consensus process.

Delegated Proof-of-Stake

DPoS is a consensus algorithm designed to enhance the efficiency and scalability of blockchain networks by addressing the limitations of traditional PoW and PoS algorithms. The concept was introduced by Daniel Larimer [11] as part of the BitShares blockchain. DPoS relies on a reputation-based voting system, where token holders vote to elect a small group of delegates (also known as witnesses) to validate transactions

and produce blocks on behalf of the network. This method reduces the number of nodes involved in the consensus process, thus significantly improving transaction throughput and reducing latency. The DPoS protocol operates in a network where the stakeholders (token holders) participate in a voting process to select delegates. These delegates are responsible for maintaining the blockchain, verifying transactions, and generating new blocks. The stakeholders hold the delegates accountable, as they can be voted out at any time if they fail to act in the network's best interest. The DPoS protocol follows a sequence of steps to reach a consensus, which ensures that the blockchain remains secure and efficient. The detail of DPoS is given below.

1. **Election of Delegates**: The first step in the DPoS process is the election of delegates. Token holders use their voting power, proportional to their stake in the network, to elect a group of trusted delegates. These delegates are responsible for block production and transaction validation. The number of elected delegates is typically much smaller than the total number of nodes in the network, which helps in achieving faster consensus.

2. **Block Production**: Once elected, delegates take turns producing blocks in a round-robin manner. Each delegate is given a time slot during which they are responsible for creating a block. This block contains the transactions that have been broadcasted to the network. If a delegate fails to produce a block within their designated time slot, the opportunity passes to the next delegate, and the missing block is left empty, ensuring that the network continues to operate smoothly.

3. **Validation and Approval**: After a block is produced, it is broadcast to the network for validation. The other delegates review the block to ensure it adheres to the network's rules and does not contain any invalid transactions. Once validated, the block is added to the blockchain. The reputation system within DPoS ensures that delegates are motivated to act honestly, as their position depends on maintaining the stakeholders' trust.

4. **Voting and Governance**: In addition to block production, DPoS includes a built-in governance mechanism where stakeholders can vote on protocol changes, system upgrades, or the replacement of underperforming delegates. This democratic process ensures that the network remains adaptable and responsive to the needs of its users.

1.3 SMART CONTRACTS

With the emergence of the Ethereum blockchain, smart contracts came into existence, which are self-executing programs written in Solidity programming language. Smart contracts run on the blockchain and are automatically triggered when certain predefined conditions are met. That is, the execution of smart contracts does not require any intermediaries. The smart contracts programs contained agreements with certain conditions. These contracts exist across a distributed and decentralized blockchain network, which enables the network to execute the trusted transactions among the anonymous participants without the need for central authority [16]. Smart contracts have enabled a wide range of applications, from automated token sales to complex decentralized applications that run on the Ethereum blockchain network. For example, Let user A (buyer) agree to buy a certain service from user B (seller) using a smart contract. The contract is written into the blockchain and might stipulate that when B provides the digital token for the car's ownership, the agreed amount of cryptocurrency from A's wallet is automatically transferred to B's account.

This exchange is transparent, where all parties can verify the transaction, and once executed, it cannot be reversed. While smart contracts offer numerous benefits, they are not without challenges. The host to execute the smart contracts is a blockchain network, which is immutable; therefore, it cannot be modified once the smart contract is executed on the blockchain. Thus, the smart contract vulnerabilities (i.e., bugs or oversights in programming) can lead to unexpected results. The vulnerable contracts lead to unwanted forks in the blockchain. Despite these challenges, there are various applications of smart contracts. One widely known smart contract application is Decentralized Autonomous Organization (DAO) [17]. DAO is governed by smart contracts, making operations independent and transparent. It is a token based governance, where decision making is distributed among the network participants according to their token/stake holding.

1.4 TYPE OF BLOCKCHAIN

Broadly, blockchain is categorized into three types: public blockchain, private blockchain, and consortium blockchain. This categorization is designed with specific use cases and how the nodes are added to the network. Below, we have briefly defined all these categories.

1.4.1 Public Blockchain

A public blockchain is a type of blockchain that allows individuals to participate and opt-out at any moment of the time. It is also called a permissionless blockchain because participating and opting out of the network does not require any permission procedure. The native cryptocurrencies and tokens associated with this blockchain are used as incentives for the participants. The participants can join the network to read, write, and participate in the consensus process. This type of blockchain is fully transparent, decentralized, and secure through a consensus process. Despite these advantages, there are multiple challenges, such as scalability, privacy, and limited performance. The widely known examples include Bitcoin, Ethereum, and Litecoin.

1.4.2 Private Blockchain

Opposite to public blockchain, a private blockchain is a type of blockchain that is operated under a controlled environment. That is, the joining and opting out requires permission; therefore, it is also called a permissioned blockchain. It is useful to organizations that want to use the benefits of blockchain's secure and efficient ledger capabilities but are afraid to put their data in the open network. In contrast to the public blockchain, the private blockchain offers fast transaction confirmation time and higher throughput due to the comparatively simple consensus algorithms that operate amongst trusted and authenticated participants. The customizable nature of these blockchains enables organizations to fine-tune them to fit their requirements. Operating within a controlled environment also means private blockchains can function with less redundancy and resource expenditure than their public blockchains, which makes them cost-effective. The widely known examples include Hyperledger Fabric [18] and Corda [19].

1.4.3 Consortium Blockchain

Consortium blockchains offer a hybrid approach that incorporates the benefits of both public and private blockchains. Selected nodes manage the network of a consortium blockchain, thus providing partial decentralization. This partial decentralization balances the tradeoff between shared control and independent operations. While it allows a higher degree of transparency and security than private blockchains, it restricts participation in the consensus process to a limited number of approved participants. Consortium blockchains are often used in sectors where multiple stakeholders need to securely and transparently share information, such as banking, healthcare, and cross-industry collaborations. A known example of a consortium blockchain is Quorum [20].

In conclusion, the selection amongst the public, private, and consortium blockchains directly depends on the requirement of specific applications and operational models (i.e., open and restricted). A public blockchain is ideal for scenarios demanding transparency and wide participation, and a private blockchain offers an optimal solution for enterprises requiring control and confidentiality. Consortium blockchain is suitable for collaborative projects among multiple stakeholders. In Table 1.1, we provide a comparative analysis between public, private, and consortium blockchains.

Table 1.1: Comparative analysis between Public, Private, and Consortium blockchains

Feature	**Public Blockchain**	**Private Blockchain**	**Consortium Blockchain**
Access	Open to anyone	Restricted, permissioned access	Restricted, permissioned, controlled by multiple organizations
Decentralization	Fully decentralized	Centralized	Partially decentralized
Governance	Community-based or algorithmic	Single organization or consortium	Multiple organizations with shared control
Speed	Slower due to consensus protocols	Faster as there are fewer nodes	Moderately fast, optimized for scale of participants
Security	High, distributed risk	Dependent on the central authority	High, with shared responsibilities
Transparency	Fully transparent	Limited transparency, known participants	Selectively transparent, known participants
Control	No single entity has control	Single organization has control	Multiple pre-selected entities share control
Consensus	PoW, PoS, etc.	Could be simplified consensus	Often more complex than private, less than public
Scalability	Challenging due to size and consensus complexity	Easier due to fewer nodes	Easier than public, more complex than private
Energy Consumption	Potentially high in PoW systems	Lower due to fewer nodes and efficient consensus	Varies, generally more efficient than public blockchains
Use Case Examples	Cryptocurrencies, global computing platforms	Supply chain, internal record-keeping	Inter-bank settlements, consortium led supply chains

1.5 CHALLENGES IN BLOCKCHAIN

Despite many advantages, there are various challenges in blockchain technology that need to be addressed, such as scalability, interoperability, and privacy. Scalability remains a significant challenge in blockchains like Bitcoin, which struggles with limited transaction throughput. Interoperability to enable cross-chain

communication between different blockchain platforms is another challenge. Additionally, the privacy of users and transactions is a big concern, specifically when blockchain is being adopted in applications such as energy trading in smart grid systems. Apart from these challenges, there are some important vulnerabilities in existing blockchain protocols. The vulnerabilities include double-spending attacks, illegal forking, Liveness, 51% attack, and 1% attack. Below, we have defined all these challenges and vulnerabilities in detail.

1.5.1 Scalability

As blockchain systems are decentralized, every transaction must be verified across potentially thousands of nodes. This process ensures security and integrity but limits transaction speed and increases costs [21]. Public blockchains like Bitcoin can handle almost 7 transactions per second, while Ethereum can process about 30. This is much less than traditional payment systems like Visa, which can handle thousands of transactions per second. Layer-2 solutions [22], sharding [23], rollups [24], and Plasma [25] are the techniques to address this issue. Layer-2 solutions, sharding, and sidechains are the prominent techniques to address the scalability issue [26]. From these three techniques, Sharding has emerged as a potential solution to the scalability challenges faced by blockchain networks. However, recent evaluations of state-of-the-art blockchain sharding protocols have revealed significant limitations and challenges, such as securing sharded networks against attacks, robust leader selection mechanisms, and high cross-shard communication overhead. In sharding, the network is divided into small groups called shards. These shards are prone to attacks (i.e., shard takeover or 1% attacks), making the whole blockchain network vulnerable to attackers. These vulnerabilities highlight the need for more secure, robust solutions to prevent attacks. Additionally, an effective leader selection mechanism is the critical component of the efficient performance of the sharding-based blockchain. The state-of-the-art schemes of leader selection in sharded blockchain are either biased or prone to attacks. Therefore, an unbiased and effective leader selection is needed to achieve the highly performing blockchain. Furthermore, high cross-shard communication overhead is another hurdle in achieving the expected outcome of the sharded blockchain. Due to this, the throughput of the blockchain decreases instead of increasing. Inappropriate shard allocation and transaction placements into shards are the root cause of this performance bottleneck.

1.5.2 Interoperability

Interoperability in blockchain systems refers to the ability of different blockchain networks to communicate, share data, and conduct transactions with one another seamlessly, i.e., enabling cross-chain communication. It is a foundational aspect that can significantly determine blockchain technology's widespread adoption and utility. However, interoperability presents various challenges, such as the presence of third-party intermediaries, different blockchain platforms, and transactional finality [27]. The presence of a third party as an intermediary between two blockchains is prone to a direct attack from the adversary, and the trust in the third party is questionable. The existing cross-chain communications are based on the third party intermediary. Designing an interoperable blockchain protocol without a third party intermediary is the biggest challenge. Apart from this, blockchains can vary significantly in their underlying protocols. They may use different consensus mechanisms (e.g., PoW, PoS, and DPoS), have varying block validation processes, and handle state management and storage differently. These variations make it challenging to achieve direct communication and transaction execution across chains. Furthermore, different blockchains have different ways of achieving transaction finality. Some may consider a transaction final after a certain number of block

confirmations (e.g., six blocks in Bitcoin), while others may use alternative mechanisms. Harmonizing these differing finality conditions is necessary for interoperability [28].

1.5.3 Privacy

Public blockchains provide an open ledger where transaction details are visible to all the participants of the network. While this ensures transparency, it also leads to potential privacy concerns. Additionally, blockchain transactions are pseudonymous, not anonymous. Though blockchain transactions do not directly reveal identities, they can often be linked to individuals or organizations through analysis or off-chain information, another cause of privacy breaches. Privacy issues magnify when blockchains are adopted in applications beyond cryptocurrencies, such as in energy trading in smart grid [29], where the exposure of personal data can have far-reaching implications. These systems often handle sensitive personal information that, if not protected properly, could result in privacy violations [30]. There are various possible solutions available to handle privacy concerns; out of them, zero-knowledge proofs (ZKP) are the best-suited solution with some appropriate modifications. The ZKP is a cryptographic principle, which is essential where privacy must be preserved, like in blockchain. ZKP is a cryptographic method where one party can prove to another that a statement is true without revealing any specific details about the statement itself. Cryptocurrencies like Zcash [31] employ ZKPs (specifically zk-SNARKs [32]) to ensure transactions are recorded, but details like sender, receiver, and transaction amount remain concealed [33].

1.5.4 Vulnerabilities in Blockchain

Despite the inherent security advantages, several vulnerability issues threaten the stability and trustworthiness of blockchain networks. Among these, some of the most critical attacks include double-spending, forking, Liveness, 51% Attack, and 1% Attack or Shard takeover attacks. Below, we describe all these vulnerabilities in detail.

1. **Double-Spending Attack:** The double-spending attack is a fundamental problem in transaction processing governed by blockchain technology. It occurs when a user manages to spend the same digital currency more than once. There are multiple ways to launch this attack; for example, an attacker or adversary manipulates the delays to broadcast the conflicting transactions, which get included in two different blocks. Although the consensus algorithms are designed to prevent this attack. However, attackers exploit the vulnerabilities in these algorithms and take advantage of them to include duplicated transactions or invalid transactions in the confirmed block. This attack is usually the resultant attack of other attacks, such as forking and Liveness, which we have defined below. The mathematical foundation for the probability of a double-spending attack succeeding could be derived based on the Poisson distribution.

   - q be the fraction of the total network hash rate controlled by the attacker,
   - p be the fraction of the total network hash rate controlled by honest nodes, with $p+q=1$,
   - z be the number of confirmations the network requires.

   The probability P that the attacker catches up from a deficit is given by:

$$P(z) = 1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \tag{1.1}$$

where λ is the expected number of blocks the attacker should have found by this point, which is $\lambda = z\frac{q}{p}$. This function describes the cumulative probability that the attacker will find at most z blocks in the period, reflecting the difficulty of the attack as the number of confirmations z increases.

2. **Forking:** Forking in blockchain usually refers to the creation of an alternative chain where two blocks have the same parent block. A fork in the chain is shown in Figure 1.5. There are two types of forks in the blockchain: soft fork and hard fork. A soft fork is a backward-compatible update in the existing blockchain protocol. It requires a majority of the network nodes to update in order to enforce new rules. Segregated Witness (SegWit) is an example of a soft fork in the Bitcoin blockchain [34], [35]. A hard fork is a disruption to the existing blockchain protocol [36]. It is a planned fork, often used to add significant new features. Bitcoin Cash (BCH) [37], [38] is an example of a hard fork in the Bitcoin blockchain, which increased the block size limit from 1 MB to 7 MB. Apart from the soft and hard forks, forking could be legitimate and illegitimate. A legitimate fork is when honest nodes update or create new features in the existing protocol, whereas an illegitimate fork occurs when malicious or unauthorized nodes change the protocol with the intent to disrupt the blockchain protocol. The objective of an illegitimate fork could be to perform a double-spending attack or create a fraudulent version of the blockchain. An illegitimate fork typically occurs when an attacker secretly mines blocks, adds them to the ledger and makes it longer than the honest chain [39].

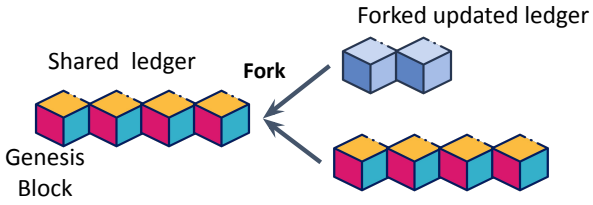


Figure 1.5: Fork in blockchain

3. **Liveness Attack:** Liveness in blockchain refers to the ability of continuous transaction processing and adding new blocks to the ledger. Therefore, a Liveness attack is said to occur when the attacker succeeds in halting the progress of the blockchain. This could happen when the attacker disrupts the network's operation by flooding it with fake transactions or blocks. The Liveness of the blockchain can also be disrupted when the attacker controls a significant portion of the network to prevent the confirmation of legitimate transactions. Let T_{normal} be the transaction confirmation time in normal conditions, and T_{attack} be the transaction confirmation time when the attack occurred. Similarly, B_{normal} and B_{attack} are the block production rates in normal and attack conditions, respectively. A simplified formula to show the impact of the Liveness attack (LA) is represented as follows:

$$LA = \alpha\left(\frac{T_{attack} - T_{normal}}{T_{normal}}\right) + \beta\left(\frac{B_{normal} - B_{attack}}{B_{normal}}\right) \tag{1.2}$$

The α and β are the weighting factors that reflect the relative importance of each metric (i.e., transaction confirmation and block production) to the blockchain's liveness. This formula gives a quantitative measure of the liveness impact, with higher values of LA indicating that the attack impact is high. During the attack, it is expected that the transaction confirmation time increases and the block production time decreases.

4. **1% Attack or Shard Takeover Attack:** The 1% attack is one of the major issues in sharding based blockchain systems. It exploits the reduced security that can occur within individual shards. In

a traditional, non-sharded blockchain, an attacker typically needs to control most of the network's hashing power (in PoW systems) or stake (in PoS systems) to compromise the system, often described as a 51% attack. Meanwhile, in a sharded blockchain, the network's total resources are equally divided among all the shards. In this scenario, controlling a single shard requires significantly less computation than controlling half of the network. For example, if a blockchain network is divided into 100 shards, an attacker might only need to control 1% of the total network to take over a single shard, hence the term "1% attack." Once a shard is compromised, the attacker could make targeted manipulations in the transactions within that shard and violate the integrity of the transactions. This vulnerability highlights the need for carefully designed security measures in sharded blockchains.

1.6 BLOCKCHAIN APPLICATIONS

In this chapter, we have discussed various aspects of blockchain technology, from its fundamental background to the inherent challenges it faces. However, this discussion would be incomplete without exploring the practical applications of this technology. Blockchain gained prominence primarily due to the success of cryptocurrencies, especially Bitcoin. As a result, blockchain is widely recognized for its role in the cryptocurrency market. However, researchers have identified its potential in many other fields, such as healthcare, smart cities, transportation, energy distribution, and more. This thesis is dedicated to three applications: the Internet of Things (IoT), Intelligent Transportation Systems (ITS), and Smart Grids. These applications are critical components of any future smart city. However, integrating blockchain technology with these applications is not as simple as using plug-and-play tools due to the existing challenges and issues associated with blockchain technology, as discussed earlier. In the envisioned smart cities of the future, blockchain's role will be pivotal in ensuring data security, integrity, and decentralization. As urban infrastructures become increasingly interconnected through IoT devices [40], [41], the enormous amount of data generated necessitates a secure and tamper-resistant storage mechanism. Blockchain's decentralized structure emerges as an ideal solution, ensuring that diverse data streams—spanning traffic control and energy distribution—are preserved with integrity and transparency [7]. Therefore, an enhanced and improved blockchain version must be investigated before integrating this technology into applications. Below, we outline the three applications for which we have proposed blockchain-based security solutions.

1.6.1 Internet of Things (IoT)

The transformative potential of IoT promises both substantial economic benefits and societal advancements [42]. It envisions an interconnected world, facilitating seamless communication and service delivery. Such advancements have been instrumental in the evolution of smart cities, especially in areas like smart homes, smart healthcare, and environmental monitoring [43]. However, the security of these interconnected systems is highly essential. Conventional security measures often fall short in IoT applications due to the high computational requirements needed to maintain data security. Moreover, IoT devices' limited computational power and energy constraints mean that traditional cryptographic methods are often impractical. Furthermore, conventional IoT application security solutions are based on a centralized architecture, which is prone to a single point of failure. Therefore, we have designed three schemes for blockchain-based security solutions in IoT applications, namely, Secure Communication and Authentication for IoT Applications using Blockchain (SCAB-IoTA), Blockchain-based Integrated Security System for Internet of Things (IoT) Applications (BLISS), and Blockchain-Enabled Security System for Internet of Things Applications (BSS).

1.6.2 Intelligent Transportation Systems (ITS)

Modern smart cities face significant challenges in traffic management in transportation systems. The statistics of global and Indian road accidents emphasize the design of efficient ITS and show why it is one of the biggest challenges in traffic management. For instance, road accidents rank as the ninth leading cause of death globally, claiming approximately 1.5 million lives annually [44]–[46]. Efficient ITS offers a solution that aims to enhance road traffic efficiency [47]. With the inclusion of IoT devices and interconnectivity, the performance of the ITS increases many folds. However, this inclusion exposes the system to potential vulnerabilities to the attacker. Blockchain offers a solution to address these vulnerabilities. Therefore, we have designed two solutions for ITS based on blockchain technology, which are a Lightweight Blockchain Security Protocol for Secure Storage and Communication in SDN-Enabled IoV (LBSV) and A Novel Incentive Mechanism for Vehicles in Intelligent Transportation System Based on Consortium Blockchain (SmartCoin).

1.6.3 Smart Grid

With the emergence of smart cities and blockchain technology, the demand for energy is also increasing. Conventional energy sources are either highly polluting, highly costly, or both. The green energy is the better alternative [48]. However, the distribution of this energy is challenging when it comes to the individual households generating it on the rooftop of the house [49]. Modern smart grids, strengthened with internet capabilities, promise dynamic energy distribution and pricing. Yet, their centralized architecture is prone to data compromises and systemic failures. Blockchain offers promising solutions to these issues. While blockchain holds the potential to secure smart grid energy trades, succeeding protocols are restricted by computational constraints and inefficiencies. Existing cryptocurrencies also struggle to transparently facilitate energy trading, with performance and security issues [29], [50]–[52]. Therefore, to address these challenges, we have designed three schemes for energy trading, which are blockchain Protocol for Secure Transaction Processing in the Next Generation Smart Grid (SmartGrid-NG), Blockchain-Based Energy Trading in the Local Energy Market (LEM) Using Modified Double Auction Protocol (ETradeChain), and Privacy-Preserved Blockchain Framework for Energy Trading in Smart Grid Using Zero Knowledge Proof (zk-SGB).

1.7 MOTIVATION AND CONTRIBUTIONS

Having established a foundational understanding of blockchain technology, its consensus mechanisms, and the diverse applications and challenges inherent in its integration across various domains, this thesis is driven by the need to address several pressing issues that hinder blockchain's broader adoption and efficacy. While blockchain offers unprecedented opportunities for decentralization, security, and transparency, significant challenges remain in areas such as scalability, interoperability, and security, particularly in complex and dynamic environments like IoT, ITS, and smart grids. This thesis aims to tackle these challenges by posing critical research questions and developing novel solutions that push the boundaries of what blockchain technology can achieve. Each chapter is dedicated to exploring these questions in depth, offering innovative contributions that advance the state-of-the-art in blockchain research and its practical applications.

The scalability of blockchain has been explored through questions such as: "*How can blockchain systems be designed to handle a growing volume of transactions without compromising decentralization and security?*", "*What novel consensus mechanisms or structural modifications can be introduced to overcome the inherent limitations of existing sharding-based solutions?*," and "*How can cross-shard communication and synchronization issues be mitigated to ensure high throughput and low latency in scalable blockchain systems?*" These questions are addressed in Chapter 3 of the thesis. Interoperability in blockchain has been investigated with questions like: "*How can different blockchain platforms be made to communicate and transact with each other without relying on third-party intermediaries?*", "*What algorithms or protocols can be developed to enable secure, trustless, and efficient cross-chain asset transfers?*," and "*How can we ensure that the fundamental characteristics of blockchain (non-repudiation, unlinkability, confidentiality, and atomicity) are preserved in an interoperable environment?*" Chapter 4 is dedicated to answering these questions with respect to the proposed scheme called CrossLedger.

Additionally, the thesis aims to strengthen the blockchain ecosystem, particularly in IoT, ITS, and smart grid applications, by exploring the following questions: "*What blockchain-based solutions can be implemented to enhance the security of IoT applications, particularly in terms of secure communication and authentication?*", "*How can ITS benefit from blockchain technology in terms of secure data dissemination and protection against common cyber-attacks like Sybil attacks, man-in-the-middle attacks, and data tampering?*", "*What are the trade-offs in integrating blockchain with smart grid systems, and how can we design a secure and efficient energy trading system that addresses these trade-offs?*," and "*How can a new cryptocurrency, tailored for energy trading in decentralized smart grids, be designed to prevent issues like unfair pricing and ensure secure transactions?*" Chapter 5 of the thesis answers all these questions for each application separately. In addressing these research questions about scalability, interoperability, and the blockchain ecosystem, the following are the major contributions of this thesis:

- **Restructured Blockchain Architecture:** The thesis introduces *BlockTree*, a novel blockchain architecture that restructures the traditional linear blockchain into a nonlinear structure, enabling parallel processing of multiple transactions. This restructured design significantly enhances scalability by allowing the blockchain to handle a higher volume of transactions without compromising decentralization or security.
- **Cross-Shard Consensus Mechanism:** The thesis presents *UnoShard*, a sharding-based solution that introduces a novel cross-shard consensus mechanism. This mechanism ensures seamless communication and synchronization between shards, reducing latency and improving throughput in scalable blockchain systems. Additionally, *UnoShard* incorporates a trust-based PBFT consensus mechanism within each shard, enhancing security and reliability in transaction validation.
- **Generalized Cross-Chain Asset Transfer Protocol:** The thesis introduces *CrossLedger*, a pioneering protocol that facilitates cross-chain asset transfers between heterogeneous blockchain platforms without relying on third-party intermediaries. *CrossLedger* employs innovative algorithms such as the Asset Forwarder Selection (AFS), Trust Establishment, and Asset Transfer and Confirmation (ATC) algorithms to ensure secure, trustless, and efficient interoperability between different blockchain networks.
- **Integrated Security System for IoT Applications:** To address the research questions, we proposed two blockchain-based schemes, *BLISS* and *SCAB-IoTA*, to enhance the security of IoT applications. *BLISS* utilizes smart contracts on the Ethereum blockchain to create secure clusters of IoT devices,

providing decentralized authentication and secure data transmission within these clusters. *SCAB-IoTA* incorporating angular distance measurement for clustering movable devices and using a hybrid cryptosystem (ECC and AES) for secure communication. Together, these schemes ensure robust security for IoT devices, particularly in dynamic and mobile environments, reducing risks such as device impersonation and data breaches.

- **Lightweight Blockchain Security Protocol and Incentive Mechanism for ITS:** The thesis introduces *LBSV* and *SmartCoin*, two innovative schemes for enhancing security and efficiency in ITS applications. *LBSV* is a lightweight blockchain security protocol that integrates a modified PBFT consensus algorithm and a hybrid SDN architecture, ensuring secure storage and communication in SDN-enabled IoV networks. *SmartCoin* complements *LBSV* by providing a novel incentive mechanism based on consortium blockchain, which promotes honest behavior among vehicles in ITS through the use of a new digital currency, SmartCoins (STC). Together, these schemes improve secure data dissemination, protect against cyber-attacks, and enhance the reliability and efficiency of the ITS environment.
- **Secure and Efficient Energy Trading in Smart Grids:** This thesis presents three blockchain-based systems namely, *SmartGrid-NG*, *ETradeChain*, and *zk-SGB*, which are designed to provide security, privacy-preservation, and optimized energy trading in decentralized smart grids. *SmartGrid-NG* integrates smart grid technology with the GridChain consensus algorithm, enhancing the security and reliability of energy transactions by eliminating vulnerabilities such as single points of failure. *ETradeChain* introduces a new energy cryptocurrency called Pcoin and employs a modified double auction and Vickrey auction methods to ensure fair and transparent energy trading. *zk-SGB* further ensured the privacy of the user in energy trading. These schemes address key challenges in smart grids, such as transaction security, unfair pricing, and efficient energy distribution, making them crucial for the development of next-generation energy systems.

1.8 THESIS STRUCTURE

Beginning with an introduction to define challenges and vulnerabilities in blockchain in **Chapter 1**, we have covered the related work in **Chapter 2**, providing context for the proposed solutions. **Chapter 3** describes the challenges and solutions to the scalability of blockchain. The proposed schemes BlockTree and UnoShard are described in this chapter to show the potential solution to improve the scalability. The BlockTree is a scalable blockchain solution with a hierarchical structure that significantly improves transaction finality, while the UnoShard is a scheme to harmonize scalability and security in blockchain sharding. **Chapter 4** is dedicated to the interoperability in the blockchain. The proposed CrossLedger scheme addresses the critical need for seamless asset transfers across different blockchain networks, marking a significant step towards blockchain interoperability. The thesis then progresses to practical applications of blockchain in IoT, ITS, and Smart Grids in **Chapter 5**. In this chapter, we first see the schemes proposed for the IoT application, which are BLISS and SCAB-IoTA. After that, we went deeper into the ITS applications and explained the two proposed schemes, LBSV and SmartCoin. After covering IoT and ITS, we explain the blockchain-based solutions for energy trading in the smart grid, which are SmartGrid-NG, ETradeChain, and zk-SGB.