IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 October 2023, accepted 5 December 2023, date of publication 12 December 2023,
date of current version 20 December 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3342079*

TOPICAL REVIEW

# Blockchain: A Crypto-Intensive Technology–A Comprehensive Review

**MUHAMMAD IMRAN SARWAR[1], (Member, IEEE), LOUAI A. MAGHRABI[2], (Member, IEEE), IMRAN KHAN[1,3], QAMAR H. NAITH[4], (Member, IEEE), AND KASHIF NISAR[1], (Member, IEEE)**

[1]Department of Computer Science & IT, Superior University, Lahore 54500, Pakistan
[2]Department of Software Engineering, College of Engineering, University of Business and Technology, Jeddah, Saudi Arabia
[3]School of Computer Science and Informatics, Cardiff University, CF10 3AT Cardiff, U.K.
[4]Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah 21959, Saudi Arabia

Corresponding author: Muhammad Imran Sarwar (info@imranchishty.com)

**ABSTRACT** Blockchain has shifted the paradigm of computer-based commercial applications during the last decade. Initially developed as a public ledger for Bitcoin transactions, it has already shown that it has the potential to revolutionize the world, where trust, security, privacy, and anonymity are the assurances. The data stored within the blockchain remains unchangeable, resistant to tampering, and distributed across multiple locations within a decentralized network. The existence and reliability of blockchain rely heavily on robust cryptographic primitives, as these are fundamental to its operation. While blockchain faces significant challenges in the ever-evolving landscape of hardware and software technologies, it has retained its reputation for being secure due to its underlying cryptographic primitives. The architecture of blockchain, various consensus protocols, and the impacts of quantum computing are also discussed here. This study reviews the existing academic literature on cryptographic primitives used in blockchain and endeavours to bridge the gaps and provide a detailed understanding of their role in blockchain security. An exploratory qualitative research methodology is used in this study and is based on the latest literature on the topics. The findings of this study provide a valuable reference to the knowledge body and enhance the comprehension of blockchain, cryptography, and cryptographic primitives in blockchain for both new and experienced researchers, enabling them to identify new opportunities and challenges in the domain.

**INDEX TERMS** Asymmetric cryptography, blockchain, consensus protocols, digital signatures, elliptic curves, hash functions, Merkle root, quantum-proof digital signatures.

## I. INTRODUCTION

Blockchain is a kind of digital distributed ledger that works on a decentralized network, ensuring security and providing immutability of the data. A blockchain network consists of interconnected nodes that validate and incorporate new transaction blocks into the existing chain. These blocks contain transactions and possess some distinct cryptographic hashes that connect them to the preceding block, forming a linked chain of blocks known as a blockchain [1]. Modifying information within a block would require altering subsequent blocks, ensuring the complete security of the data. The security of blockchain is a crucial aspect and it relies on the cryptographic primitives to guarantee the authenticity and integrity of recorded transactions [2]. This characteristic sets blockchain apart as a superior technology. Blockchain possesses essential features such as decentralized data distribution across a network of nodes, the near-immutability and permanence of data stored in blocks, and transparent operations. All network nodes have access to an identical copy of the distributed ledger, and any changes or updates are visible to all participants. Originally developed as a platform for the cryptocurrency Bitcoin, blockchain aimed to create a decentralized, public infrastructure capable of preserving data privacy, security, integrity, and anonymity. However, after demonstrating its potential in Bitcoin, blockchain has been gaining attention in other domains. Sectors including finance, accounting, healthcare,

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen.

supply chain management, education, IoT, and public sector governance are areas where blockchain-based applications have already been implemented [3].

Over the past decade, blockchain has undeniably transformed the landscape of computer applications. Since its emergence in 2008, blockchain has proven its capacity to address vital needs in global record-keeping systems [4]. The true identity of the blockchain and Bitcoin's creator(s) remains unknown. Yet their development was intended to establish a secure transaction mechanism while safeguarding user anonymity [5]. The transaction process in blockchain eliminated the necessity for intermediaries, and there was no requirement for a pre-existing trust relationship between the parties involved [6]. In the blockchain network, each node perceives itself as the custodian of the data, enabling the system to operate without any central authority [7]. A blockchain is a chain of blocks and each block contains a timestamp, Nonce, nBit, Merkle Root hash, the hash value of the previous block, and a set of transactions.

The demand for blockchain-based applications is rapidly increasing. It is expected that blockchain will generate $2 trillion in revenue by 2030 [8]. It is important to distinguish between the technology and its implementations, such as Bitcoin and Ethereum. Various cryptocurrency applications, including Bitcoin, use the blockchain as a foundational technology. It operates as a database for storing records or data, with Bitcoin using the blockchain as its data storage solution. Blockchain is a combination of cryptographic primitives and works on decentralized peer-to-peer networks [9]. The significance of blockchain lies in its ability to allow participants to operate and govern the system without the need for intermediaries [10]. It functions as a public ledger not owned by any specific entity, providing a trustworthy and reliable platform for managing transactions. Blockchain-based solutions can address vulnerabilities commonly found in Artificial Intelligence (AI) and the IoT [11]. By using robust cryptographic hash functions, blocks in the blockchain are interconnected [12]. Every block in the blockchain has the hash value of its previous block.

Consequently, any modifications made to the data within a previous block will change its hash value, altering the hash value of all subsequent blocks in the chain. It makes it extremely difficult, if not impossible, to modify the contents of any block without detection [13]. The immutability and security of the blockchain stem from this linking process. Any attempt to alter a previous block would require the attacker to recalculate the hashes of all subsequent blocks since each block is dependent on the one preceding it, so it becomes computationally infeasible to modify the contents.

### A. RESEARCH METHODOLOGY

The objective of this study is to present a comprehensive review of the latest research on cryptographic primitives used in blockchain. The findings of the study contribute to enhancing the comprehension of blockchain, cryptography, and cryptographic primitives in blockchain. This study carried out a structured approach to search, evaluate, and review the most up-to-date and relevant studies. An exploratory qualitative analysis methodology is used in this study and is based on the latest academic literature. It reviews some of the latest studies on a variety of topics related to blockchain, cryptographic primitives, SHA, RIPMED, and related fields.

#### 1) LITERATURE SELECTION

The literature searched was from reputable databases, including IEEE, MDPI, Taylor & Francis, Google Scholar, ACM, Science Direct, ResearchGate, and Springer Link, with a particular emphasis on the most relevant publications in recent years. Additionally, student theses were also considered and evaluated. In our literature searches, we employed specific search terms such as ''blockchain security and integrity'', ''cryptographic primitives'', ''secure hash algorithms'', security and privacy in blockchain'', ''elliptic curve'', ''blockchain distributed security architecture'', ''digital signatures'', ''blockchain consensus protocols'', ''quantum computing and blockchain'', and ''security vulnerabilities in blockchain''. The search was conducted and a total of 780 papers were retrieved.

#### 2) INCLUSION AND EXCLUSION CRITERIA

The criteria for inclusion and exclusion of studies were also set to help refine the search results further. The criteria for inclusion and exclusion are as follows:

Inclusion Criteria: The studies specifically examined cryptographic primitives in the context of blockchain. Studies were published in reputable journals and conference proceedings.

Exclusion Criteria: Papers that were out of the scope of this study. Grey literature from blogs and websites. Irrelevant, unpublished, and low-quality studies. Papers that were more than 5 years old. Non-English publications and duplicate articles were also not included.

#### 3) SELECTION RESULTS

The literature selection process was undertaken following four steps: 1: identification, 2: screening, 3: eligibility, and 4: grouped/included. At first, 780 studies were identified from various sources: 747 from databases and 33 from other sources. Out of those, 312 studies were removed before the screening process as they were deemed irrelevant or did not cover the topics fully. In the next step, 468 studies were screened and analyzed to determine if they fulfilled the requirements. After analyzing their titles, abstracts, and keywords, a further 203 studies were excluded in the second step. In the third step, the eligibility of the literature was checked by applying inclusion and exclusion criteria to 265 studies for further refinement. 166 studies were excluded: 14 were found to be out of scope, 18 were published in other than the English language, 93 studies were found to be of low quality or unpublished, 34 duplicate studies were found, and 7 papers were not found. The study publishing period criteria were

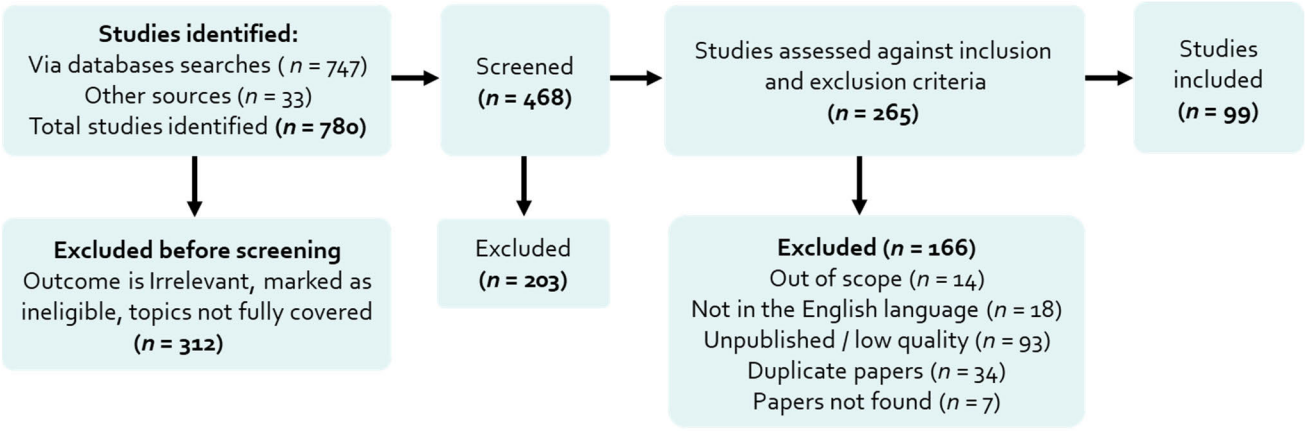



**FIGURE 1.** Literature selection process.

relaxed because the number of studies found after filtering was insufficient.

Finally, 99 studies were left after the filtering process, and they are included in this study. Figure 1 shows the literature selection process for this study.

### B. STRUCTURE OF THE ARTICLE

The remainder of the paper is organized as follows: Section II presents the background of blockchain, the structure of blockchain, block structure, types of blockchains, P2P network structure, and the Bitcoin transaction process. The cryptographic primitives, hash function, digital signature, and Merkle root are explored in Sections III, IV, and V, respectively. Consensus protocols, their types and workings, and a comparative analysis of popular consensus protocols are presented in Section VI. The discussion part is covered in Section VII before conclusions are drawn in the final section.

## II. BACKGROUND OF THE STUDY

### A. BLOCKCHAIN AT A GLANCE

The roots of blockchain technology can be traced back to various critical milestones in its innovation history. In 1982, Chaum introduced protocols as the basis for blockchain technology [14]. Subsequently, in 1991, Haber and Stornetta defined cryptographically coupled block chaining [15]. Merkle trees were then introduced in 1993 [16]. In 1998, Szabo presented Bit-Gold, a digital currency that operated on a decentralized network [17]. The breakthrough moment came in 2008 when Nakamoto introduced Bitcoin, a cryptocurrency that used an electronic cash transaction mechanism on a decentralized peer-to-peer network [18]. During this time, the term blockchain gained recognition as the underlying technology supporting Bitcoin [19]. Over time, blockchain technology continued to mature, leading to the development of new variants and approaches. In 2013, Buterin introduced Ethereum, which brought about significant advancements [20]. In 2015, the Linux Foundation introduced Hyperledger Fabric, an open-source, permissioned version of the blockchain framework [21]. Through these historical developments and subsequent innovations, blockchain technology has evolved, offering diverse applications and capabilities [22]. The distinctive features and characteristics of blockchain set it apart from other technologies in the science and technology domains [23]. Through the utilization of a cryptographic hash function, blocks within a blockchain are connected and linked together. Each block contains a header that holds the hash value of the preceding block's header, forming a continuous chain of interconnected blocks known as a blockchain. When a new block is created, it includes a reference to the header hash of the previous block. Any modifications made to the data in a previous block will result in an alteration of its hash value, which, in turn, will change the hash value of every subsequent block in the chain.

Consequently, attempting to change the contents of any block without detection becomes extremely difficult. The immutability and security of the blockchain derive from this linking process. Altering a previous block would require the attacker to recompute the hashes of all subsequent blocks since each block relies on the information of the preceding one. This computational challenge makes it virtually impossible to modify the contents of the blockchain without being noticed, rendering it a secure and trustworthy method for managing and storing data [24]. Initially, the development of blockchain was solely aimed at facilitating Bitcoin transactions and ensuring data security. There was no intention to utilize blockchain in any other domain. However, its potential to address confidentiality and immutability has become evident in recent years, leading to its application in various vital areas and industries beyond cryptocurrency [25]. Blockchain operates on a fully decentralized peer-to-peer network without any specific entity, user, company, or group governing or controlling its operation [26]. In addition to its use in cryptocurrency, blockchain has found relevance in diverse domains. These include applications related to security and privacy control [27], healthcare [28], copyrights, energy, advertisements, supply chain [29], automobiles [30], agricultural production,

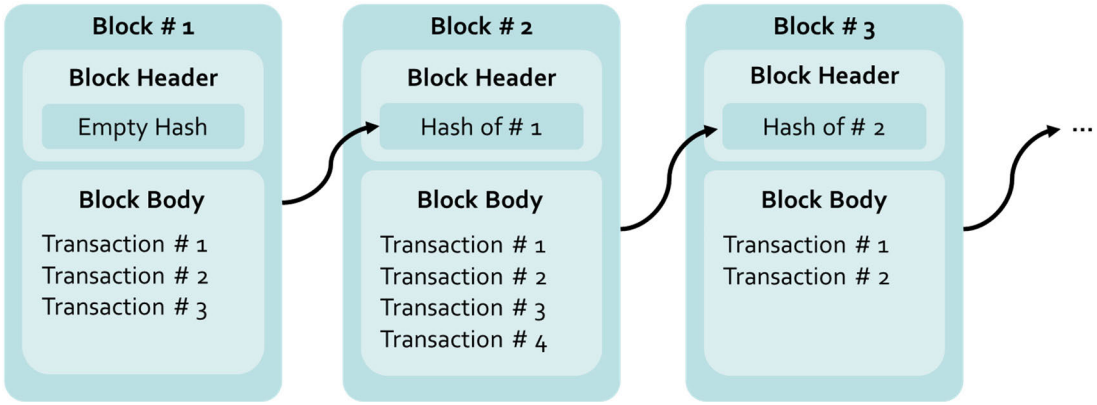



**FIGURE 2.** Structure of the blockchain.

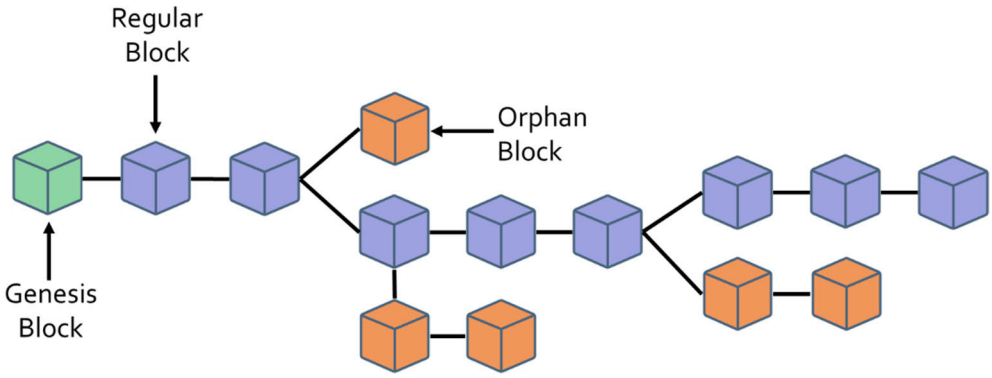



**FIGURE 3.** Formation of the blockchain.

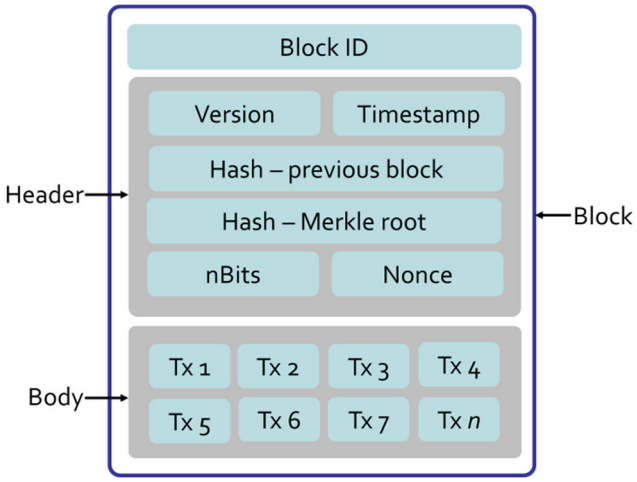



**FIGURE 4.** A Simplified Bitcoin block.

electronic health records (HER) management applications [31], customer and citizen identity management [22], IoT [32], and intrusion and fraud detection [33]. Blockchain was introduced in 2008 to provide a platform for recording Bitcoin transactions without the involvement of third parties. This concept of eliminating intermediaries from the system was groundbreaking and revolutionary, as it had never been attempted before.

### *B. STRUCTURE OF BLOCKCHAIN*

In the blockchain, the main or original chain is determined by considering only the longest chain, while any blocks outside this chain are known as orphan blocks. Orphan blocks are generated when miners attempt to mine and add blocks without authorization, resulting in their inability to persist and grow like the original main chain. Figure 2 illustrates the blockchain as a sequence of interconnected blocks tightly linked using SHA, a cryptographic hash function. Each block holds the hash value of its previous block. The initial block in the chain is called the genesis block and it does not have any hash value because it is the starting point of the chain [34]. Figure 3 depicts the formation of the blockchain.

#### 1) BLOCK STRUCTURE

A block is a fundamental component of blockchains and is divided into the header and the body, as shown in Figure 4.

The header consists of information that is mainly related to the block and does not include any transactions or transaction

components. Typically, each block contains the following information: [35], [36]

1) Block ID: Each block is assigned a unique identifier for identification purposes.
2) Version: A 4-byte value indicating the version number of the block, which can be helpful for future system or software upgrades.
3) Timestamp: A timestamp in seconds obtained from the UNIX Time Server as a 4-byte numeric value is added in the header of every block.
4) Previous Block Hash: A 32-byte, 64-character hash value representing the previous block's hash value.
5) Merkel Root: A 32-byte hash value derived from all transactions in the block's body.
6) nBits: A 4-byte value indicating the computational difficulty level for block mining.
7) Nonce: A 4-byte random number used in the computation process to generate a correct hash value. Nonce stands for "Number Only Used Once," indicating that each nonce value is used only once.

The blocks within a blockchain serve as the fundamental data structure, housing the transactions within them. These transactions are permanently recorded and cannot be modified, deleted, or further added once the block has been added or mined into the main chain. In the case of Bitcoin, transactions are consolidated into blocks, and each block contains one or more transactions. Miners are responsible for adding or mining blocks into the chain, and upon successfully forming a block, miners receive a reward in the form of bitcoins. This reward transaction, a coinbase transaction, is included in the list of transactions within the block's body [37].

Figure 5 provides some additional details on the coinbase transaction and other transactions recorded in the body of a block.

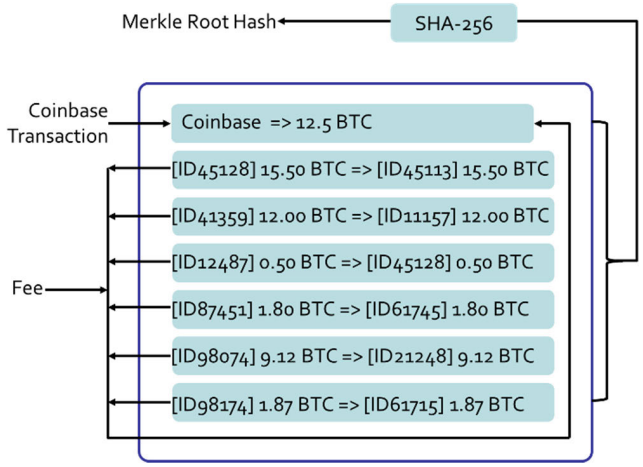



**FIGURE 5.** Bitcoin block's body structure.

### 2) BITCOIN TRANSACTION PROCESS

There are various steps involved in a Bitcoin transaction and Figure 6 depicts the flow of a Bitcoin transaction.

A Bitcoin transaction is completed in various steps [38], which are as follows:

1) Initiating a Transaction: When a user intends to transfer bitcoins to another user or recipient, they commence the transaction process. They use a dedicated Bitcoin application called a wallet for this purpose [28]. The transaction includes essential details such as the recipient's Bitcoin address, the amount of bitcoin being sent, and other relevant information. The sender digitally signs the transaction using a private key. This digital signature proves that the sender authorized the transaction and ensured its integrity and security.
2) Broadcasting the Transaction: After the transaction has been signed, the user's wallet sends the transaction message across the Bitcoin network. It transmits this transaction information to multiple computers, known as Bitcoin nodes, and they are connected to the network.
3) Propagating the Transaction: The broadcasted transaction spreads throughout the network, eventually reaching miners after passing through various nodes. Miners, participants within the Bitcoin network, are responsible for validating and adding transactions to blocks.
4) Verifying the Transaction: Miners thoroughly examine various aspects to determine the validity of a transaction. They ensure that the transaction adheres to the network's rules and protocols, possesses a valid signature and that the sender has sufficient funds to cover the transaction's cost.
5) Inclusion in a Block: Once a miner confirms the transaction's validity, they include it in a candidate block. Miners engage in a competitive process of solving a challenging mathematical puzzle known as Proof-of-Work (PoW) to add a new block to the blockchain.
6) Block Confirmation: The transaction is confirmed if a miner solves the puzzle. Then, the miner adds the newly created block to the blockchain. A confirmed transaction becomes part of the transaction history and is permanently recorded on the blockchain.
7) Transaction Finality: As additional blocks are added to the blockchain, the transaction becomes more secure and resistant to tampering. The possibility of a transaction being reversed decreases with each subsequent block added to the chain. After adding six blocks to the chain, Bitcoin transactions are deemed final and irreversible.

### 3) NETWORK STRUCTURE

Blockchain works on a decentralized network where multiple nodes or participants collaborate to collectively uphold and verify the blockchain without relying on a central entity. No single organization or central server possesses complete control or power over the system within a decentralized blockchain network. Figure 7 illustrates a P2P decentralized network in which all nodes are connected.

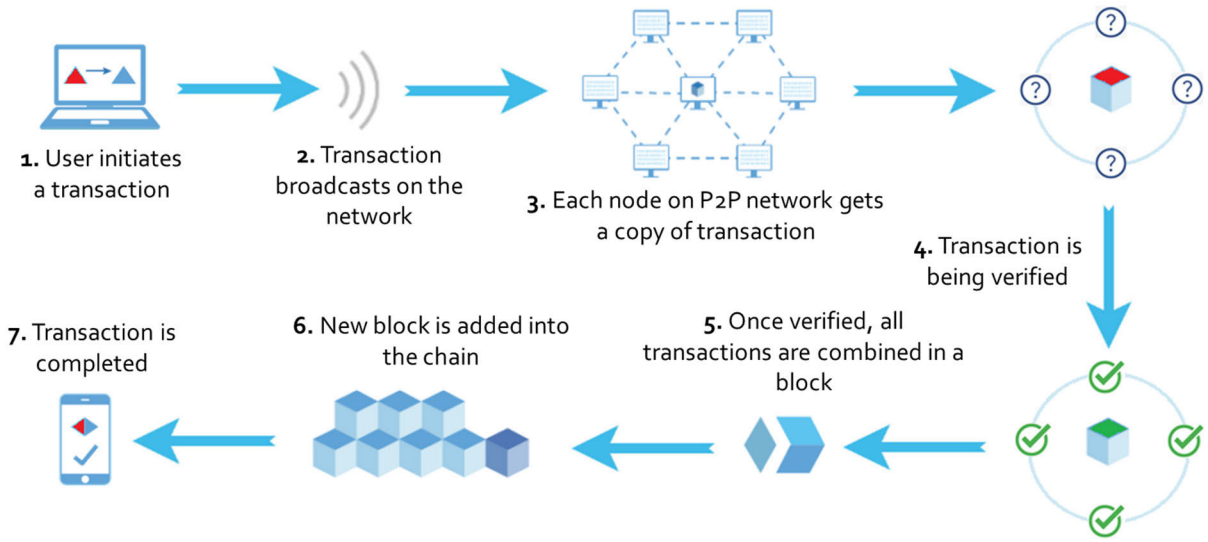



**FIGURE 6.** Bitcoin transaction process.

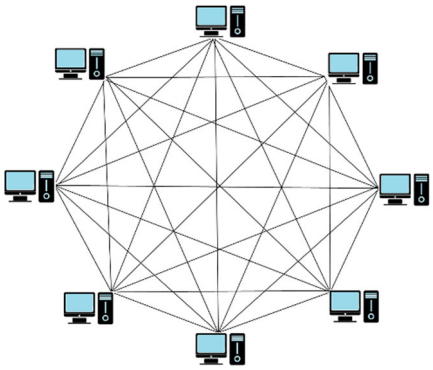


**FIGURE 7.** P2P decentralized network.

Decentralization in blockchain networks offers numerous benefits such as enhanced security, transparency, and robustness. Below is a summary of how a decentralized blockchain network functions:

1) Peer-to-peer (P2P) network: The blockchain network operates on a peer-to-peer network structure, where multiple nodes or computers directly communicate. Nodes within the blockchain possess a complete copy of the entire ledger.
2) Transaction Verification: On the blockchain network, nodes validate and verify transactions, ensuring their authenticity and compliance with predefined consensus rules.
3) Decentralized Governance: As blockchain governs in a decentralized manner, decision-making processes are based on consensus, where participants vote on the authenticity and validity of transactions, block mining, etc.
4) Distributed Ledger: The blockchain functions as a distributed ledger, wherein participating nodes possess a copy of the complete transactions. Whenever a new transition occurs, it is communicated to the entire network.
5) Block Creation and Addition: Once transactions are verified, they are organized into blocks and added sequentially to the blockchain. This process creates a chain of interconnected blocks, each containing a reference to the preceding block.
6) Consensus Mechanism: Nodes participate in a consensus mechanism to validate and agree upon the state of the blockchain. This mechanism ensures that all participating nodes reach a consensus regarding the legitimacy of transactions and the order in which they are appended to the blockchain.

#### 4) CONSENSUS MECHANISM

Bitcoin operates on a public blockchain, and a copy of all transactions is available on each full node. When a user initiates a transaction, it is broadcasted on the network and is added to a transaction pool in an unconfirmed state. The pool transactions are then published on the network and are visible to every full node, and they verify each transaction. Once the transactions are verified, they become available to add to the public ledger or blockchain [39]. The transactions are combined into a block published on an append-only blockchain [40]. This process is called block mining. Miners also combine their resources to get more computational power and resources, which is called the miners' pool. A PoW is a consensus algorithm that allows miners to agree on adding a block to the blockchain. In the algorithm, the miners have to compute the value of Nonce to complete a 256-bit hash value. Computing the puzzle requires powerful computers for computation [41]. A few miners solve a puzzle of low difficulty, and vice versa. Once a block is added to the blockchain,

it is considered confirmed. The consensus achieved by the network makes it exceptionally computationally intensive to reverse, thus rendering transactions in confirmed blocks as final and irreversible. The probability of finding Nonce for PoW, $H$ for a given target $T$ is defined in Equation (1) [42].

$$\mathrm{P(H \leq T) = T/2^{256}} \tag{1}$$

### C. TYPES OF BLOCKCHAIN

The general classification divides blockchains into three main categories according to their features. There is another category, which is a hybrid model of blockchain that is a combination of public and private blockchains [43]. Figure 8 illustrates various types of blockchains.

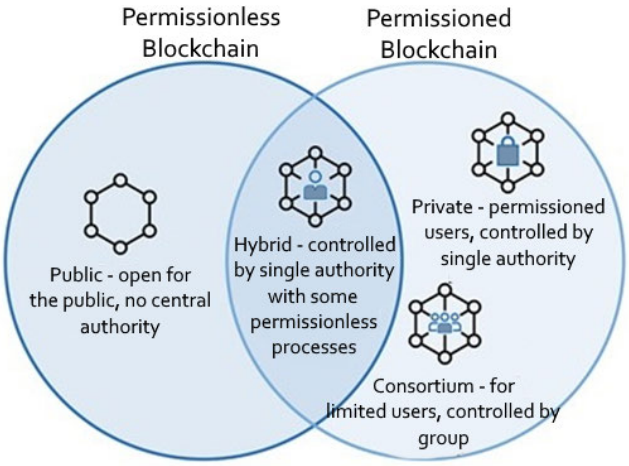



**FIGURE 8.** Various types of blockchains.

#### 1) PUBLIC OR PERMISSIONLESS BLOCKCHAIN

In a public or permissionless blockchain, permission to enter the network is unnecessary. It operates as an open and publicly accessible system without a centralized authority. Users have unrestricted access to a public blockchain that is not under the ownership or control of any person or organization [44]. In this particular kind of blockchain, each node in the network operates as an equal participant and retains a copy of the entire transaction in the blockchain. A public blockchain allows anyone to engage in the process of reaching a consensus and authenticating transactions. Bitcoin and Ethereum are prominent examples of public blockchains. Consensus protocols like PoW, Proof-of-Stake (PoS), and Delegated Proof-of-Stake (DPoS) are employed in public blockchains [45].

#### 2) PRIVATE OR PERMISSIONED BLOCKCHAIN

A private or permissioned blockchain allows access only to authorized users. It operates as a closed and secure system, offering higher security and privacy than a public blockchain [44]. A single authority or a designated group governs the network. A private blockchain is employed and maintaining privacy and security are of utmost importance, as specific nodes and users are allowed to conduct transactions. This type of blockchain prioritizes speed, scalability, and security as its main features. It finds extensive use in supply chain management and facilitating intercompany transactions within a federated environment. In a private blockchain, Practical Byzantine Fault Tolerance (PBFT), Proof-of-Authority (PoA), and Proof-of-Elapsed-Time (PoET) consensus protocols are used [45].

#### 3) CONSORTIUM BLOCKCHAIN

The consortium blockchain is a blend of public and private blockchains. In this type of blockchain, a consortium comprising multiple organizations or entities collaboratively manages and operates it. A consortium blockchain is particularly beneficial when several organizations require cooperation while simultaneously upholding certain levels of privacy and control. In a private blockchain, PBFT, Proof-of-Vote (PoV), and Proof-of-Trust (PoT) consensus protocols are used [45].

#### 4) HYBRID BLOCKCHAIN

The hybrid blockchain model combines the best parts of both public and private blockchains. Private transactions can happen within a network that is connected to a public blockchain to make things safer and clearer [46]. Users interact with the network through smart contracts [47]. This arrangement allows for the selective sharing of information while leveraging the decentralized nature of the public blockchain. The blockchain is still immutable while under control, making changes or deletions impossible. The network uses automated smart contracts, and roles and permissions are predefined. Table 1 summarizes various types of blockchains, their advantages and disadvantages, and the cases in which they are used.

### D. BLOCKCHAIN VULNERABILITIES

#### 1) 51% ATTACK

A 51% attack, also known as a majority attack, is a potential threat in which the attacker gains control of more than half of the network in order to disrupt it. If a single attacker or a group of people pool their resources and gain control of more than 50% of the total network, then they can override the network's consensus mechanism and carry out malicious activities such as double-spending. An attacker can execute a 51% attack when it has enough mining power to intentionally change the order of transactions, preventing some or all transactions from being confirmed. This situation is known as transaction denial of service. If the attacker or attackers gain access to the network, they can also prevent or stop the other miners from performing mining activities, resulting in a mining monopoly.

#### 2) DDOS

Due to blockchain's digital nature, it is vulnerable to attack and exploitation. DDoS attacks on a blockchain concentrate

**TABLE 1.** Types of blockchain and their features.

| | Public (Permissionless) | Private (Permissioned) | Consortium | Hybrid |
|---|---|---|---|---|
| Advantages | It works independently, ensures transparency, and is highly trustable | Provides better control, privacy, high performance | Better access control, secure, scalability | Better access control, high performance, secure, scalable |
| Disadvantages | Performance and scalability issues due to open access, less secure | Not much trustworthy, Auditability | Transparency issues | Privacy can be compromised, and technical issues when upgrading |
| Use-case | Mainly used in cryptocurrencies, document management and verification | Supply chain management, federated environment, assets management | Insurance and banking, federated environment, supply chain | Electronic health record-keeping, real estate |

on the protocol layer, with transaction flooding posing the greatest threat to blockchains. Traditional DDoS attacks can be executed against a blockchain to slow down its operations, and attackers can use the blockchain ecosystem to execute DDoS attacks. The majority of blockchains have a fixed block size and limit the number of transactions per block. By sending spam transactions to the blockchain, attackers can fill up the blocks with fake transactions and prevent the addition of legitimate transactions. All legitimate transactions will then be stored in the mempool, awaiting the next block. With the proliferation of blockchain applications, a new type of denial-of-service (DoS) attack has emerged: the blockchain denial-of-service (BDoS) attack. These assaults target blockchains like Bitcoin and utilize the proof-of-work (PoW) consensus mechanism.

#### 3) ECLIPSE ATTACKS

The attacker overshadows the target node's connection, which is why it is called an eclipse attack. An eclipse attack on the blockchain is a specific type of attack that is not very common. When the attack does take place, the attacker isolates a particular node on the P2P network, prevents it from accessing the network, and starts directly interacting with that node. All incoming and outgoing traffic from that affected node is redirected to the attacker's nodes.

### *E. CRYPTOGRAPHIC PRIMITIVES IN BLOCKCHAIN*

In blockchain, strong cryptographic functions were used that changed the entire scenario, and now we have a technique for storing data that can be trusted [48]. Blockchain is a type of ledger that contains transactions and their details, such as date, time, amount, sender, and receiver. The transactions are recorded in blocks linked to each other and form a chain of blocks. The transaction recorded in the ledger becomes permanent and cannot be deleted or altered [49]. The blocks are distributed to many nodes on a P2P network; therefore, multiple copies of the data are available for validation [50]. At the core of blockchain is its cryptographic primitives, which make it valuable. There is no concept of any third party in a blockchain, and it works in a fully decentralized manner. Therefore, security on the blockchain is of the utmost importance. Cryptographic primitives are used to secure networks [40]. They are the fundamental algorithms used to create higher-level security in blockchain and serve as the cryptosystem's core. Securing information and communications so only the intended recipient can decode and process them is known as cryptography [51].

Information access by unauthorized parties is prevented. Cryptography combines the words "crypt," which means hidden, and "graphy," which means writing. The techniques used in cryptography are based on mathematics, and these techniques or algorithms ensure that messages are transmitted in ways that make them difficult for any unauthorized person to decode. These algorithms are also used for creating cryptographic keys and digital signatures, protecting privacy, and securing private information on internet transactions like purchases and payments. The fundamental building blocks to maintaining a high level of security in a blockchain are cryptographic primitives [52]. They are, therefore, considered the most critical parts of blockchain for security, integrity, confidentiality, encryption, decryption, authentication, and validation.

Cryptography is frequently associated with encoding plain text with the intention that only the recipient can decode it. This process is known as encryption, while decryption converts the encrypted text into plain text so the original message can be read. Some key features of cryptography are listed as follows:

1) Confidentiality: The information is highly secure, confidential, and only for the intended person. Cryptography ensures confidentiality by encrypting data so only the intended recipient can decrypt it. It is done by converting the data into an unreadable format using a key. The only person with the key can change the data back to what was initially sent to them.
2) Integrity: Cryptography ensures the integrity of the data using the hash function, which creates a distinct fingerprint or hash value for each piece of data. Any modification or deletion of any part of the data produces a changed hash value, which makes it easy to spot any data manipulation. Information cannot be altered or deleted in storage or during transmission.
3) Non-repudiation: Cryptography ensures non-repudiation by ensuring that the sender cannot claim that they never sent the data (i.e. the sender or originator cannot deny that they sent the data) and for this purpose

**TABLE 2.** Cryptographic primitives in various cryptocurrencies [40].

| Mainstream Cryptocurrencies | Hashing | | | | | | Digital Signature | | | | | | Consensus | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SHA-256 | Ethash | Scrypt | X11/X13 | Equihash | RIPEMD160 | ECDSA | EdDSA | Ring | One-Time | Borromean | Multi-Signature | PoW | PoS | RPCA | BFT | Others |
| Ark [53] | ✓ | | | | | | ✓ | | | | | ✓ | | | | | ✓ |
| Bitcoin [18] | ✓ | | | | | | ✓ | | | | | ✓ | ✓ | | | | |
| Binance Coin [54] | ✓ | | | | | | ✓ | | | | | | | ✓ | | | |
| Bitcoin Cash [55] | ✓ | | | | | | ✓ | | | | | | ✓ | | | | |
| Cardano [56] | | | | | | | | ✓ | | | | | | ✓ | | | |
| Dash [57] | ✓ | | | ✓ | | | ✓ | | | | | ✓ | ✓ | | | | |
| Dogecoin [58] | ✓ | | ✓ | ✓ | | | ✓ | | | | | | ✓ | | | | |
| Electroneum [59] | | ✓ | Keccak,blake256 | | | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | | | ✓ |
| Ethereum [20] | ✓ | ✓ | | | | ✓ | ✓ | | | | | | | ✓ | | | |
| IOTA [60] | | | Curl, Keccak384 | | | | | | | ✓ | | ✓ | | | | | ✓ |
| Komodo [61] | ✓ | | | | ✓ | | ✓ | | | ✓ | | ✓ | | | | | ✓ |
| Litecoin [62] | ✓ | | ✓ | | | ✓ | ✓ | | | | | ✓ | ✓ | | | | |
| Ripple [63] | ✓ | | | | | ✓ | ✓ | | | | | ✓ | | | ✓ | | |
| Solana [64] | ✓ | | | | | | | | | | | | | | | | ✓ |
| Stellar [65] | ✓ | | | | | | | ✓ | | | | | | | | | ✓ |
| Tether [66] | | | SHA-384 | | | | ✓ | | | | | | | | | ✓ | |
| XRP [63] | | | SHA-512 | | | | ✓ | | | | | | | | | | ✓ |
| Zcash [67] | ✓ | | | | ✓ | | | | | ✓ | | | ✓ | | | | |
| Zcoin [68] | ✓ | | | | | | ✓ | | | | | ✓ | ✓ | | | | |
| ZILLIQA [69] | | ✓ | | | | | EC- Schnorr | | | | ✓ | | ✓ | | | ✓ | |

**TABLE 3.** Requirements of hash function [19], [72], [73].

| Requirements | Description |
|---|---|
| Input - Variable length | $H^{*}$ can be applied to any size of data. There is no restriction on input length. |
| Output – Fixed length | The output $(h^{**})$ of $H$ must be fixed in length. |
| Efficiency | Computing $H(x^{***})$ for any given value of x could be implemented through hardware or software. |
| Preimage-resistance | Computing the value of $h=H(x)$ as $H(x)=h$ must be computationally infeasible. It must satisfy the preimage resistance requirement and ensure one-wayness. |
| Second preimage-resistance | Computing $y^{***} ! x$ with $H(y) = H(x)$ must be computationally infeasible to find for any given value of $x$. |
| Collision-resistant | Finding the pair of $(x, y)$ from $H(x) = H(y)$ must be computationally infeasible. |
| Pseudo-randomness | The output of $H$ must be statistically random. |

* HASH function, ** input of HASH function, *** output of Hash Function

digital signatures are used, giving evidence of the sender's identity and preventing them from denying that they sent the data.

4) Authenticity: The sender and receiver of the information are authenticated. Cryptography provides authentication by using digital signatures, which enable the recipient to confirm that the intended sender has sent the data. It is done by signing the data with a private key, which can be validated using the sender's public key.
5) Secure Communication: Cryptography does not allow anyone to enter the communication channel except the sender and the receiver. Therefore, communication challenges under cryptography are considered significantly restricted. Communication between the two parties takes place in well-secured ways, and any unauthorized person is strictly prohibited from entering or even intercepting the communication.

Table 2 summarizes the cryptographic primitives and consensus algorithms used in various mainstream cryptocurrencies.

## III. HASH FUNCTION

A hash refers to a mathematical function that takes an input of varying length and generates a fixed-size output called a hash value or message digest [70]. A hash function creates a distinct, irreversible representation of the input data as its output. Since hash functions are intended to be one-way operations, deducing the input data from the hash value is challenging. Hash functions are helpful in cryptography because they possess several significant characteristics. A hash function

will consistently produce the same hash value when given the same input data, quickly and efficiently, even for significantly larger inputs [71]. Finding two different inputs that produce the same hash value or finding the input data that produces a particular hash value is computationally impossible. The National Institute of Standards and Technology (NIST) released information about the SHA-2 in 2002. As per their standards, the cryptographic hash function must satisfy certain requirements as presented in Table 3.

The hash function computes the fixed-length hash value of any input in a fraction of the time, but computing an arbitrary-length input from that compact hash value is impossible [74]. Cryptographic hashing is a complex and highly sophisticated mathematical function that takes a single input and generates a fixed-length output. The mathematical complexity of SHA makes blockchain secure, and any reverse calculation to find the input value from a given output hash value is a mathematical trapdoor [38]. The input length does not matter at all. SHA is a popular cryptographic hashing algorithm with variants SHA-0, SHA-1, SHA-2, and SHA-3. It was developed by the United States National Security Agency (NSA) [75]. Bitcoin uses the SHA-2 family variant and, in particular, SHA-256. This cryptographic hash function creates an output of fixed length from an indefinite input length. Table 4 shows some essential characteristics of the SHA-2 variant family.

**TABLE 4. Characteristics of SHA-2 variants [75].**

| Features | SHA-1 | SHA-224 | SHA-256 | SHA-284 | SHA-512 |
|---|---|---|---|---|---|
| Message (input) | $<2^{64}$ | $<2^{64}$ | $<2^{64}$ | $<2^{128}$ | $<2^{128}$ |
| Block size | 512 | 512 | 512 | 1024 | 1024 |
| Word size | 32 | 32 | 32 | 64 | 64 |
| MD output bits | 160 | 224 | 256 | 384 | 512 |
| Number of rounds | 80 | 60 | 64 | 80 | 80 |

Bitcoin uses the SHA-256 and RIPEMD-160 algorithms for hashing [76]. These algorithms are used in combination, so they not only increase security but also maintain the highest level of privacy in Bitcoin transactions. The combination aims to ensure two main requirements: one-wayness and collision resistance [77]. The purpose of one-wayness is to ensure that the hash value is not reversible or, at the very least, extremely challenging to reverse, and it also aims to be collision-resistant, meaning it avoids producing the same output for different inputs. In the case of a hash value with a 256-bit output length, breaking its one-wayness would require a brute force attack 2256 times, and preventing collisions would take approximately 2256/2 attempts using a birthday attack. The SHA-256 cryptographic algorithm consistently generates the same output for a given input and satisfies properties like preimage-resistance, second-preimage-resistance, and collision-resistance. This cryptographic technique is distinct from encryption and decryption algorithms. Another cryptographic function employed in Bitcoin is RIPEMD160, a hash function based on the Merkle-Damgård construction. Bitcoin combines SHA-256 and RIPEMD160, collectively referred to as HASH160. This algorithm is used to hash transactions within a block, construct a Merkle tree, and hash the transaction signed by the sender [78]. The SHA-256 variant is more powerful and robust than SHA-0 and SHA-1.

The SHA-256 algorithm processes an arbitrary input length in the following order: to complete the length in multiples of 512 bits, the input message M is padded with 1 and then led by 0. The 64-bit tail length L of the message M stores the message's original length. The message M is then divided into 512-bit blocks as M1, M2,…Mn. Each block processes 64 rounds and generates a 256-bit partial hash as H1, H2,…Hn. This partial hash is further used for processing in the subsequent block. This process ends at the Mn block, and a 256-bit string Hn is computed as the output.

### A. ETHASH

The Ethereum platform uses a memory-hard hash function resistant to ASICs (Application Specific Integrated Circuits), specialized hardware devices for mining specific cryptocurrencies. Ethash is a heavy-duty and specialized proof-of-work mining algorithm that requires miners to exert some computational effort, generate hashes, validate transactions, and add them to the blockchain. Developing ASICs capable of generating hashes at high speeds is more challenging, resulting in a more decentralized mining process on the Ethereum network. Ethash, derived from Keccak256 and Keccak512, is considered an ASIC-resistant hash function that is used in Ethereum and other ETH-based cryptocurrencies like Expanse, MOAC, and Pirl. The highest average hash rate was reported in May 2022, which was 1,126,674 GH/s [79].

### B. SCRYPT

Several cryptocurrencies, including Tenebrix, Fairbrix, Dogecoin, and Litecoin, utilize SCrypt, a memory-hard hash function. Scrypt is designed to require more memory than Bitcoin in the long run. Innosilicon has developed Scrypt-based A12+ LTC Master Miners with a hash rate of 12.5 GH/s [80]. Scrypt was explicitly designed to counter attacks with sophisticated hardware. The A12+ LTC Master challenges the development of such specialized hardware, which can produce many hashes. Due to their memory-intensive nature, Scrypt-based cryptocurrencies are considered more decentralized and less conducive to the concentration of power among a few dominant miners [81].

### C. X11

The X11 function is a sequence of 11 different hashing operations, and it is designed to resist hardware optimization, making it challenging for specialized hardware like ASICs. Consequently, X11-based cryptocurrencies tend to be more decentralized and less influenced by a few dominant miners. The hash rate of X11 is around 227.9600 THash/s [82].

### D. EQUIHASH

Zcash, a cryptocurrency, uses an Equihash-based PoW hash function. It is designed based on the Generalized Birthday

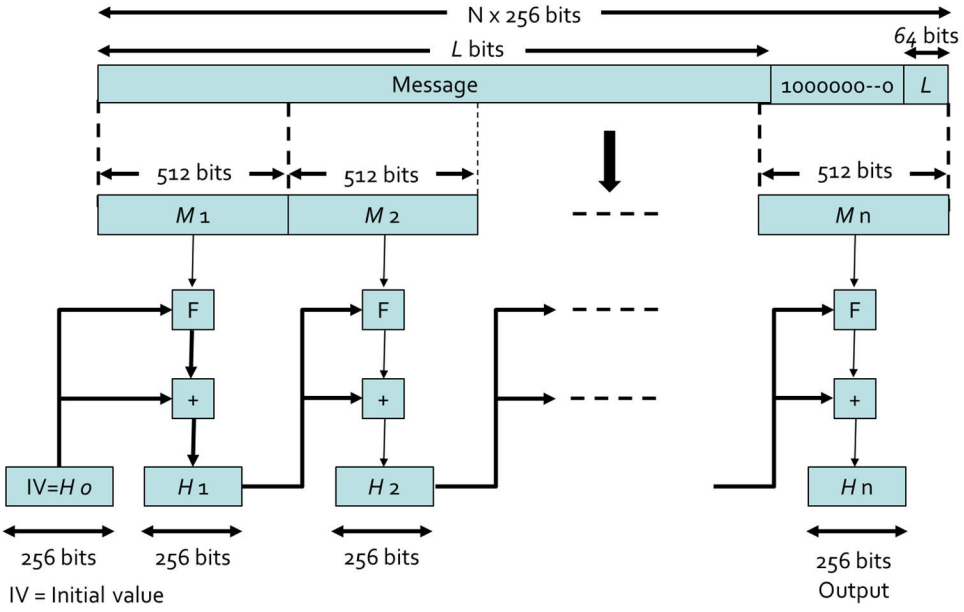



**FIGURE 9.** A conceptual view of the SHA-256 algorithm's working.

Problem, which assesses the probability of finding collisions within extensive data sets. The Equihash function is intentionally resistant to hardware optimization and can produce a considerable number of hashes in a second. This characteristic promotes a more decentralized mining process within the Zcash network and its hash-generating capacity is 917.4213 MH/s [83].

### E. ASIC-RESISTANCE

ASICs are a particular type of integrated circuit that are designed to perform specific tasks in computing. ASIC devices are specially designed to help miners in the mining process [40]. These devices are capable of producing billions of hash values in a second, which helps the miner solve the complex puzzle to mine the block [49]. There are a few cryptocurrencies like Ethereum, Monero, Haven Protocol, and Ethereum Classis that have been working and developing ASIC-resistant techniques and algorithms so that any specialized device brings no benefits to the miners.

Hash functions are known for their intensive computational design, which can affect their complexity. The computational complexity of a hash function is typically O(n), where n is the size of the input. Hence, the required hash computation time is proportional to the input size. In addition, the efficiency of the hash function depends on the algorithm used and resources such as hardware. Figure 9 depicts the conceptual view of the SHA-256 function and Figure 10 illustrates the use of cryptographic hash functions in various parts of the Bitcoin blockchain.

## IV. DIGITAL SIGNATURE

### A. ASYMMETRIC CRYPTOGRAPHY

Asymmetric key cryptography, also known as public-key cryptography, uses a pair of keys for encryption and decryption [83], [85]. The sender encrypts the information using the recipient's public key, and the recipient uses their private key to decrypt it. It is widely used for encryption and decryption. Although it is slower than symmetric cryptography, it does not require key distribution between the sender and the receiver. In this instance, one key is public, and the other is private [70]. The public key is used to lock or encrypt data. Only the receiver has access to the private key. To 'unlock' the data, the recipient uses a private key. Everyone has access to the public key, but it only functions in one way. The private key, which functions in one way and is used to decrypt the message, is in the receiver's possession. DSA and RSA are two examples of public-key cryptography.

### B. ELLIPTIC CURVE

In 1985 [86], Neal Koblitz [78] and Victor Miller developed elliptic curve cryptography (ECC). ECC uses the elliptic curve to create public and private keys linked with strong mathematical functions and asymmetric public-key encryption. A 256-bit ECC public key is equally secure compared to RSA, which generates a 3072-bit public key; therefore, the ECC-generated public key is 12 times shorter, reducing the processing time [86]. The recommended parameters for secp256k1 for a sextuple $T = \{p, a, b, G, n, h\}$ where the finite field can be found with Equation (2).

$$y^2 = x^3 + ax + b \tag{2}$$

from (2), where $a = 0$ and $b = 7$, we have:

$$y^2 = x^3 + 7 \tag{3}$$

An illustration of the elliptic curve from (3) is also shown in Figure 11.

The Standard of Efficient Cryptography Group (SECG) is an international consortium that sets standards for the ECC.

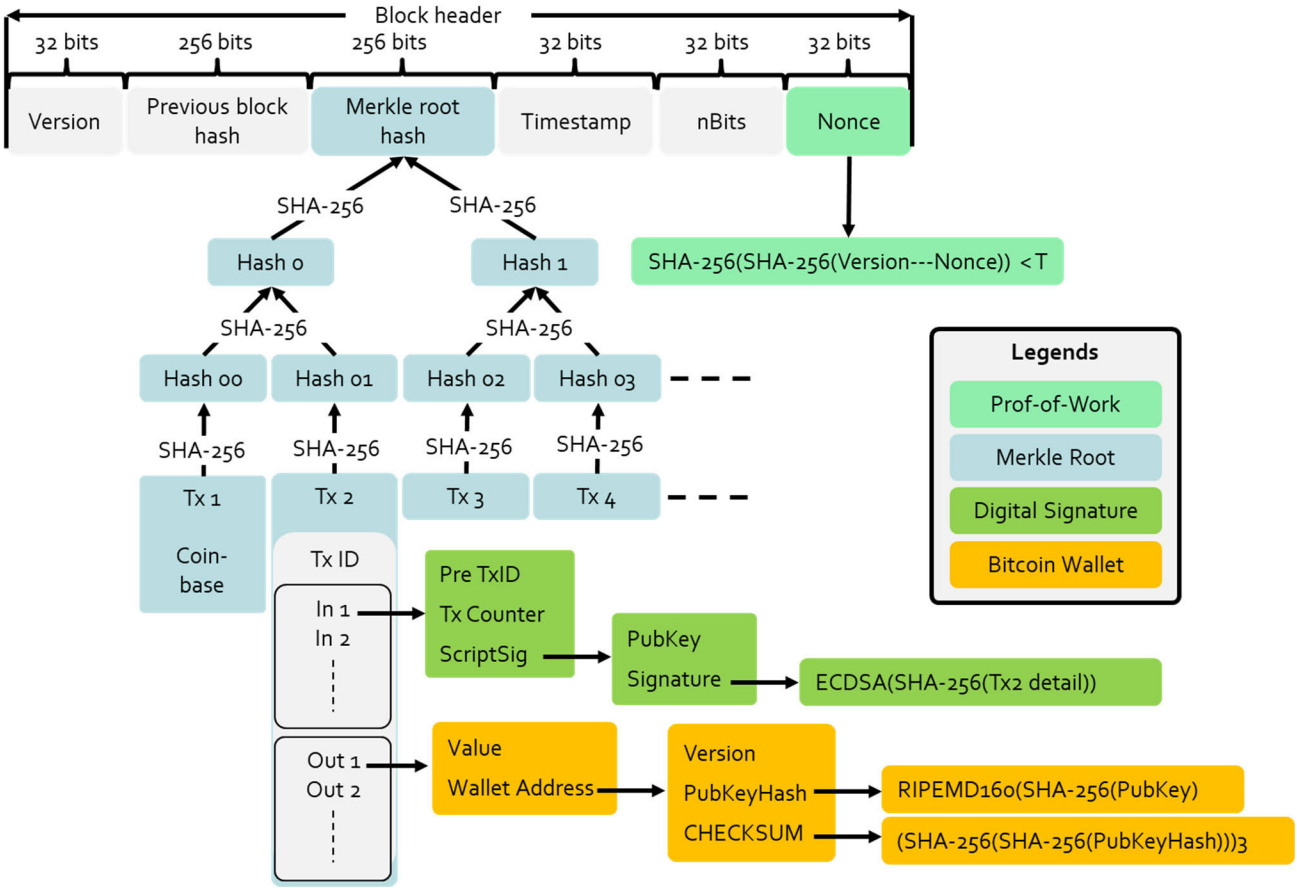



**FIGURE 10.** Cryptographic hash function in Bitcoin blockchain [40].

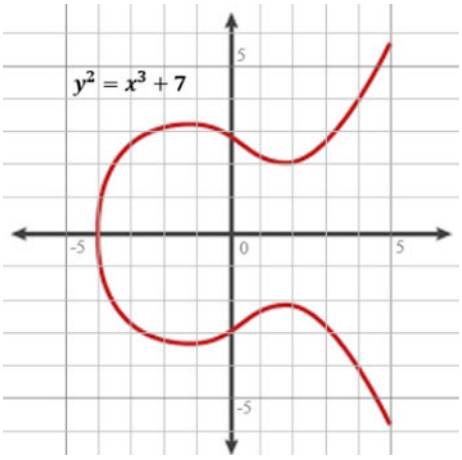



**FIGURE 11.** An illustration of the elliptic curve from (3).

The SECG has published a document recommending a set of parameters for the ECC called Elliptic Curve Domain Parameters. Secp256k1 are the parameters of ECC, which is used in Bitcoin. The algorithm used in secp256k1 relies on the Elliptic Curve Discrete Logarithm Problem (ECDLP) for mathematical complexity [87].

The ECC employs elliptic curves over finite fields, with p being a prime number and field size p = 2m. It means the field is a square matrix of p x p size, and the integer coordinates are limited to being only within the field. So, the elliptic curve equation for a finite field takes on a new form, as shown in Equation (4):

$$y^2 = x^3 + ax + b (\text{mod } p) \tag{4}$$

for secp256k1 (where a = 0 and b = 7):

$$y^2 = x^3 + 7 (\text{mod } p) \tag{5}$$

The elliptic curve over the finite field $\mathbb{F}_{17}$ would create the points as shown in Figure 12 for the following (6):

$$y^2 \equiv x^3 + 7 (\text{mod } 17) \tag{6}$$

For the calculation of elliptic curves over finite fields, (7) must always be true:

$$x^3 + 7 - y^2 \equiv 0 \,(\text{mod } 17) \tag{7}$$

To further understand this calculation, we explain it with an example. We have two points and want to know whether they exist on the elliptic curve for P = {5, 8} and P = {9, 15}. By putting the values in (7), we can find the answer:

$$\text{for P} = \{5, 8\} = (5^3 + 7 - 8^2) \bmod 17 == 0$$
$$\text{for P} = \{9, 15) = (9^3 + 7 - 15^2) \bmod 17 \,!= 0$$

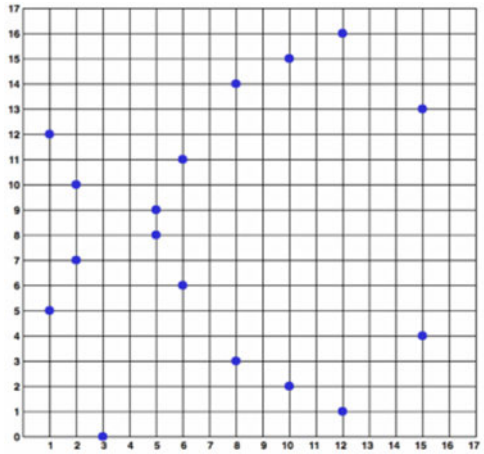


**FIGURE 12.** Point of elliptic curve for (6).

Public-key Cryptography is the underlying technology for wallets and transactions. A pair of public and private keys are generated when a user creates a wallet and becomes part of the Bitcoin network. The public key derives from the private key and is further used in the verification process. A public key is a user ID for identification, becomes the address of the user's wallet, and is publicly visible on the network. At the same time, the private key is only known to the user, which is the only proof of wallet ownership. If a private key is lost, everything associated with it is lost forever. The calculation of points on the elliptic curve is also shown in Figure 13.

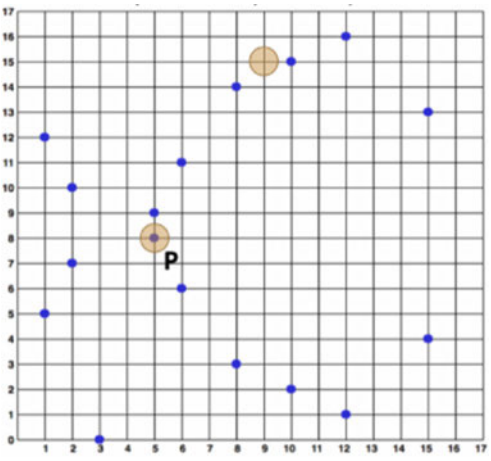



**FIGURE 13.** Calculation of points on the elliptic curve.

There are three main components of asymmetric are:

1) Private Key: The private key is a positive integer value exclusively known to the individual who generates it. By possessing a valid private key, a user can access and utilize Bitcoins by sending them or using them for transactions.
2) Public Key: Derived from the private key, the public key serves as an address for individuals in the Bitcoin system. It functions as an identifier for the holder and does not require confidentiality. The public key is publicly visible and used to verify the identity of the person initiating a transaction.
3) Signature: This mathematical value is based on the transaction and the sender's private key.

Suppose Bob wants to send a message to Alice, so he would generate a message, *S_Message*, using the original message *M* and his Private-Key, as explained in Equation (8).

$$\text{S\_Message} = \text{Sign}\,(\text{Message}\,(\text{M}), \text{Bob}'\text{s Private} - \text{Key}) \tag{8}$$

where *Sign* is a function to encrypt the message, which takes two parameters: the original message *M* and his Private-Key, then *S_Message* would transmit to Alice. Alice would verify the *S_Message* using Bob's Public-Key and the original message *M,* as shown in Equation (9):

$$\text{Result(T/F)} = \text{Verify(S\_Message, Bob}'\text{s Public} - \text{Key}, M) \tag{9}$$

Verify is a function that uses *M, S_Message,* and Bob's public key for verification and returns true or false results. Figure 14 explains the message-sending and verification processes.

### 1) ECDSA (ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM)

Public-key cryptography is a high-level framework that is used in many applications. ECDSA is used by Bitcoin to digitally sign transactions and ensure that the funds are in their rightful hands. ECDSA is an essential element of cryptosystem security that derives its security from the computational infeasibility of the ECDLP, which is hard to break. And Bitcoin uses it to ensure security and authenticity [88]. ECDSA has become the standard in the U.S. government for implementation in blockchain applications. Most blockchain-based applications generate digital signatures using ECDSA or large integer factorization problems like RSA (Rivest, Shamir, and Adleman). These algorithms work based on the computational complexity of mathematical problems [89]. The algorithms used in ECDSA have undergone deep cryptographic analysis and are considered the most secure, effective, and efficient compared to RSA or DSA [78]. A public key can be produced with the help of a simple function in ECDSA [69]. Getting a private key from that public key requires solving a mathematical discrete logarithm problem that is nearly impossible or practically infeasible [77]. Bitcoin uses the secp256k1 variant of ECC for generating public keys. Key lengths typically range from 256 to 521 bits, with 256 bits being the most common. ECDSA is known for its signature generation and verification efficiency, making it suitable for resource-constrained environments. Deterministic signatures are unavailable by default, but variants like RFC 6979 provide a standardized approach. The private key is an integer value, and the public key is a corresponding point of the private key on the elliptic curve (EC point) over finite fields [41]. Private key storage is crucial to maintaining security.

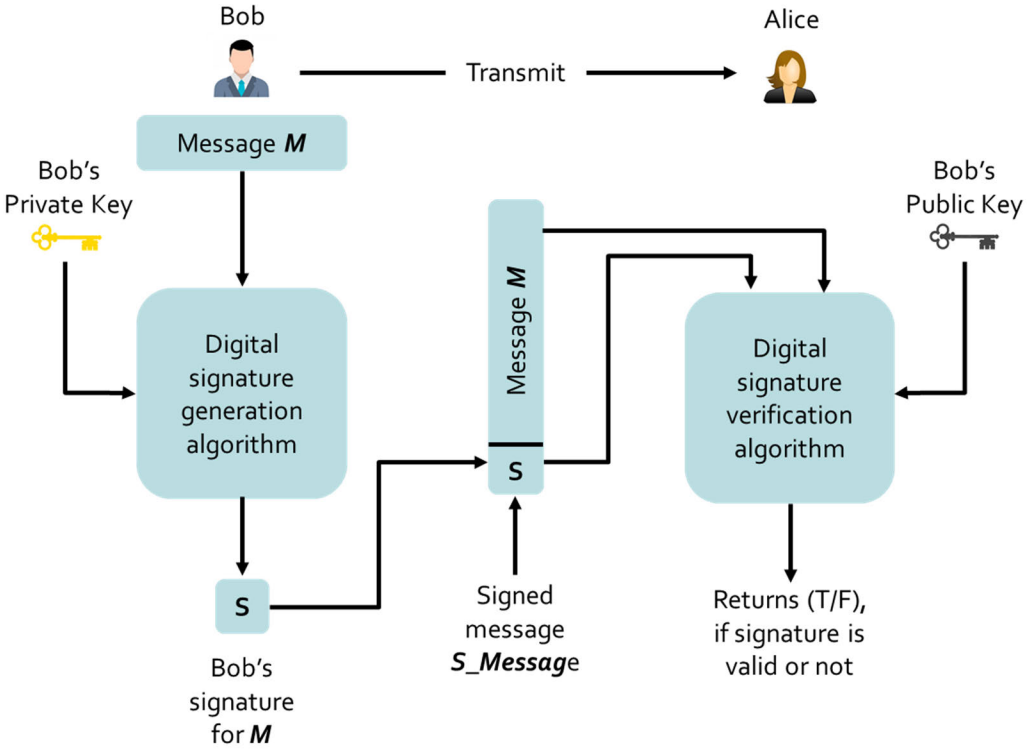



**FIGURE 14.** Message sending and verification process.

The mathematical basis for signing with ECC:
Message hash: $h = hash(msg)$
$k = h + PvtKy$ (message hash + Private key)
Generate a random number $r : (1 \ldots..n^{-1})$
Calculate random point $R = k * G$ and coordinate: $r = R.x$
Signature proof: $s = k^{-1} * (h + r + PvtKy) \pmod n$
Modular inverse: $\mathrm{k}^{-1} \pmod n = k * k^{-1} \equiv 1 \pmod n$
Return: *signature* $\{r = s\}$
The mathematical basis of verification with ECC:
Message hash: $h = hash(msg)$
Modular reverse of signature proof: $s1 = s^{-1} \pmod n$
Random point: $R' = (h * s1) * G \ (r * s1) * PubKy$
Taking the coordinate of $R' : \mathrm{r}' = \mathrm{R}'.\mathrm{x}$
Comparing if: *verify* $(r, r')$

### 2) EDDSA (EDWARDS-CURVE DIGITAL SIGNATURE ALGORITHM)

The EdDSA digital signature technique uses ECC [90] and offers secure and efficient digital signatures while aiming to reduce complexity and potential risks. It is widely regarded as an advanced option to RSA and DSA, providing robust security with shorter key lengths [91]. It offers protection against various attacks and is particularly suitable for resource-constrained environments [92]. The algorithm streamlines cryptographic operations for signature creation and verification. With its deterministic approach, EdDSA ensures consistent signatures for the same message and private key, facilitating aggregation and deterministic wallets. It has gained significant popularity in cryptographic applications, including cryptocurrencies like Monero, which uses the Ed25519 variant. Libraries support EdDSA well, making integration easier for developers.

### 3) ONE-TIME SIGNATURE (OTS)

OTS is a cryptographic method used to sign a single message or document. OTS ensures security by preventing reuse. It draws inspiration from one-time pad encryption, where each message has its own unique key. OTS generates a different key pair for each signed message. The signer creates a private and public key specifically for that message. The private key produces a secure and unique digital signature using complex mathematical operations. The recipient can verify the signature using the public key. After verification, the private and public keys associated with the signature are destroyed. OTS protects against forgery and impersonation, as each signature is unique [93]. The absence of a stored private key makes future compromises impossible. Compromising one key does not affect the security of other signed messages, ensuring perfect forward secrecy. OTS is well-suited for time-sensitive transactions or secure communication protocols, particularly in cases of unreliable online connectivity. It reduces reliance on network connectivity by allowing offline key generation and signature creation.

### 4) MULTI-SIGNATURE

Multi-signature authentication involves multiple parties signing a document or transaction, enhancing security and preventing individual control. A private key signs while the

corresponding public key verifies, ensuring authentication, integrity, and non-repudiation [94]. Multiple private keys are required for verification in multi-signature schemes, protecting against single-point failures and malicious activities. Multiple keys are needed to forge a signature, even if one is compromised. Multi-signature allows multiple parties to control a document or transaction, which is useful for consensus and involving multiple parties [95]. It creates an audit trail and makes denial of involvement challenging. Signature requirements can be customized by implementing techniques like threshold cryptography, secret sharing, and multi-party computation to distribute the signing process and avoid complete key possession.

#### 5) QUANTUM-PROOF DIGITAL SIGNATURES

Quantum-proof digital signatures are cryptographic signature schemes designed to withstand quantum computer attacks [96]. The advent of quantum computers poses a potential risk to various cryptographic algorithms, including those commonly used for digital signatures, potentially compromising their security. To address this concern, researchers have been actively exploring and developing cryptographic algorithms that can resist attacks from quantum computers [97]. These post-quantum signature schemes aim to provide long-term security, even in the face of powerful quantum computers. Quantum-proof digital signatures typically rely on mathematical problems expected to be computationally difficult, even for quantum computers.

These algorithms are built upon mathematical constructs such as lattice-based cryptography, code-based cryptography, multivariate cryptography, hash-based cryptography, and other resilient techniques against quantum attacks. The primary objective of quantum-proof digital signatures is to ensure that signatures generated using these post-quantum schemes remain secure, even if quantum computers become capable of breaking current cryptographic algorithms. Several types of quantum-proof digital signature schemes are being proposed and studied, and some of them are:

1) Lattice-based: To make encryption more secure, lattice-based cryptography schemes use the difficulty of certain mathematical problems associated with lattices and the correct way to arrange points in a space that has more than one dimension. It is considered a quantum-proof digital signature technique and a promising alternative to RSA and DSA [98]. Lattice-based cryptography facilitates public-key encryption, digital signatures, and key exchange and is used for authentication, secure communication, and data integrity. It produces larger key sizes with higher computational requirements, which somehow raises performance issues [99]. It provides improved security because lattices are very difficult to break as compared to other mathematical-based cryptography techniques. The computational process of lattice-based encryption is much faster, requires less energy consumption, and is relatively easy and flexible to implement [100]. NTRU (N-th degree truncated polynomial ring) and BLISS (bimodal lattice signature scheme) are some lattice-based schemes.
2) Code-based Cryptography: Signature schemes based on code-based cryptography, such as the McEliece signature scheme, utilize error-correcting codes. The security of these schemes relies on the difficulty of decoding linear error-correcting codes and offers protection against quantum attacks.
3) Multivariate Cryptography: There are public-key cryptography schemes that use multivariate quadratic (MQ) polynomials in which the trapdoor one-way function takes the form of an MQ polynomial map over a finite field. Multivariate cryptography-based schemes are considered to be amongst the top quantum-proof schemes [97]. They are based on MQ problems on a finite field of q elements, m quadratic polynomials $p1,\ldots, pm \in [X1,\ldots,Xn]$ in n variables. The quadratic polynomial problem is proved to be nondeterministic polynomial or NP-complete, and it seems that using quantum computers does not provide any advantage in attempting to solve it. However, NP-completeness does not exclude the possibility that for certain polynomials $p1,\ldots pm$, it can efficiently find a solution [96]. The decryption process of an encrypted message without knowing the secret key is one example of the MQ problem, and it is difficult even for a quantum computer. The special structure of $p1,\ldots pm$, may lead to some vulnerabilities, but, on the other hand, it is necessary to adopt it. Some examples of this category include Rainbow, HFE (hidden field equations), and UOV (unbalanced oil and vinegar) signature schemes.
4) Hash-based Cryptography: The Merkle signature scheme and one-time signature are some examples of hash-based signature schemes. They provide robust security by leveraging the collision resistance property of the underlying hash function and they are considered to be quantum-proof.

Some of the key features of various digital signature algorithms used in blockchain are summarized in Table 5 below:

#### 6) ECC KEY FEATURES

ECC is one of the key components used in modern cryptography due to some of its unique features and because it provides far better security than RSA and DSA. ECC's security features mainly rely on elliptic curves' mathematical complexity. This complexity makes it more secure and very difficult to solve. ECC is used in the key exchange process, digital signature and facilitates secure communication. Some of the key features of ECC are listed below:

1) Mathematical Complexity: Elliptic curves are used in ECC over a finite field. This makes it more robust as it is harder for attackers to solve the discrete logarithm problem.

**TABLE 5.** Key features of various digital signature algorithms used in blockchain.

| Features | ECDSA | EdDSA | One-Time Signature | Multi-Signature |
|---|---|---|---|---|
| Key length | 256 or 521 bits | 256 or 448 bits | Variable | Variable |
| Security | Strong (known attacks) | Strong (known attacks) | Strong (forgery) | Strong (forgery) |
| Deterministic signature | No | Yes | No | No |
| Key generation efficiency | Moderate | Fast | Fast | Moderate |
| Signature generation efficiency | Moderate | Fast | Fast | Moderate |
| Signature verification efficiency | Moderate | Fast | Fast | Moderate |
| Key management | Required secure key storage | Required secure key storage | Key destruction | Key destruction |
| Private Key reusability | Yes | Yes | No | Yes |
| Flexibility | Limited | Limited | Single-use only | Customizable threshold |
| Mostly used for | General | General | Specific | Distributed authorization |

2) Key Exchange: Elliptic Curves Diffie-Hellman (ECDH) is used with ECC, which makes communication secure on the open channels where the data is comparatively more vulnerable. The shared secret key is very challenging for reverse calculation.
3) Digital Signature: ECDSA, a well-known digital signature algorithm, also employs ECC, which is used for data integrity and authentication while still being computationally effective and hard to reverse.
4) Efficiency: ECC provides a high level of security and is more effective as it produces small key sizes, which are suitable for mobile and IoT devices.
5) Strong Resistance to Quantum Attacks: As compared to some traditional cryptography techniques like RSA and DSA, ECC seems to have quantum resilience.

#### 7) PUBLIC KEY MANAGEMENT: CHALLENGES AND MITIGATION STRATEGIES

Public key management is one of the greatest concerns in asymmetric cryptography. The user would not know that the key had been hacked or compromised until it was used by the hacker. Some of the main challenges involved in public key management are:

1) Week Keys: A week key, in terms of its length, can easily be compromised because fewer computational requirements are necessary to solve it.
2) Improper Storage: Sometimes, a user stores their key along with the data on the same computer or server, which makes it highly vulnerable.
3) Non-destruction: In some cases, the key remains valid for a certain period, and during this time, it can be used multiple times, even if it has already been used.
4) Lack of Resilience: Sometimes, the user does not get the key when it is required or is lost accidentally or due to some technical or non-technical problems. The data associated with that particular key may also be inaccessible or lost permanently.
5) Incorrect Use: Sometimes keys are generated without considering the number of times they can be used or without defining their life span. In such cases, the required security level may not be achieved.

Several solutions are suggested to help with the challenges associated with managing keys. These include keeping track of the key's life cycle, making sure that it rotates automatically, and setting up a safe and automated system for handing out keys. The key must be destroyed after its expiration date. Policy-based controls for the issuance and use of keys may also prevent their misuse and reuse.

Digital signatures' computational complexity varies and depends on the algorithm used. For example, RSA and DSA's complexity depends on the key size: RSA's complexity is roughly $O(n^2)$, where n is the number of bits in the key, while ECDSA's complexity is around O(n). In addition, the digital signature's efficiency depends on the key size. The larger the key, the more computationally expensive it is. On the other hand, with the advancement of hardware and software optimizations, the process would be more efficient.

## V. MERKLE ROOT

Merkle trees are a data structure extensively used in computer science and cryptography to effectively verify the integrity of data sets. In 1979, Ralph Merkle introduced them, hence their name. In a Merkle tree, non-leaf nodes represent the hash of their child nodes, while leaf nodes represent individual data elements [85]. The root node of the tree represents the hash of the entire data set. The key advantage of employing a Merkle tree is its ability to quickly verify the accuracy of large data sets. By knowing just the hash of the data and a few other specific hashes within the Merkle tree, a user can confirm the integrity of a particular data element. Even when dealing with extensive data sets, the user can ensure data integrity by traversing the tree from the leaf node containing the data to the root node and verifying hashes at each level. Merkle trees are commonly used in blockchain technology to verify the accuracy of transactions [71].

In a blockchain, every block contains a list of transactions, and the block header, including the Merkle Root, holds significant importance in Bitcoin and blockchain technology as it accurately represents all transactions within a specific block [101]. It is obtained from a hierarchical data structure called a Merkle tree, which organizes and categorizes transactions. When a block is created in the Bitcoin network, it consists

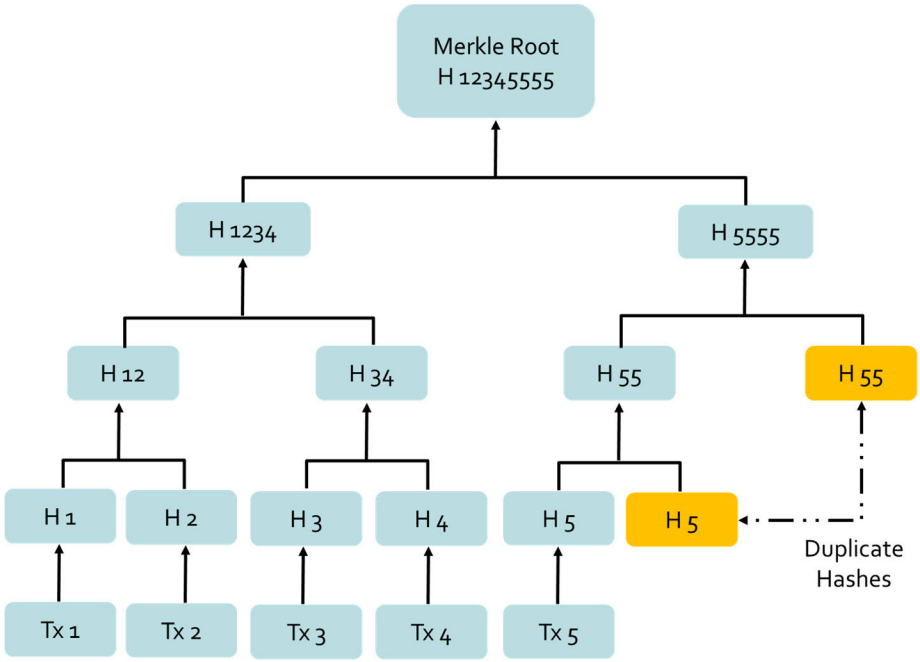



**FIGURE 15.** Formation of the Merkle tree and Merkle root.

of a collection of transactions. To store and verify these transactions efficiently, they are organized in a binary tree structure known as a Merkle tree. Each transaction is individually hashed using SHA-256 [102]. These transaction hashes serve as the leaf nodes of the Merkle tree. The transaction hash values are paired, and each pair is collectively hashed to generate a new hash, and it continues until the final hash, the Merkle Root, is obtained [103].

A Merkle tree is a complete binary tree and uses the hash value associated with each node of the tree. Merkle trees are constructed in such a way that the authentication path of a leaf can be checked against a publicly known root value [85]. A Merkle tree hash is constructed from these transactions, which allows users to quickly validate whether a transaction is included in a specific block without downloading and verifying the entire block [101]. Merkle binary hash trees derive their security from two properties of hash functions:

1) Pre-image Resistance: Computing the value of h=H(x) as H(x)=h must be computationally infeasible. It must satisfy the preimage resistance requirement and ensure one-wayness.
2) Collision resistance: Finding the pair of (x, y) from H(x) = H(y) must be computationally infeasible.

There is a recursive process in Merkle Roots that combines hashes in a binary tree structure. The number of items in the tree determines how deep the tree would be. Merkle trees have a computational complexity of O(log n), where n is the number of data items in the tree. In addition, Merkle trees are an efficient means of meeting one of the security requirements: integrity. It allows integrity verification of data without the need to compute the entire Merkle root.

Figure 15 depicts the formation of the complete Merkle Tree.

## VI. CONSENSUS PROTOCOLS

Blocks within the blockchain are appended through a process called consensus, which miners carry out via mining activity. In 1982, Lamport introduced the Byzantine Generals problem, which laid the foundation for developing consensus algorithms that address the challenges of Byzantine faults [104]. In a decentralized system like blockchain, where no central authority exists, multiple nodes collaborate to reach a consensus. Blockchain relies on a consensus mechanism to resolve disputes related to the validity of previous transactions and maintain the integrity and accuracy of the distributed ledger. Consensus represents the process by which a group of participants collectively establishes a shared understanding of the real state of the blockchain [105]. A typical mining process involves the following steps:

1) Collecting Transactions: To incorporate a set of transactions into a block, miners collect them from different sources, like their own wallet, the Bitcoin network's mempool, where unconfirmed transactions are stored, and a mining pool.
2) Verifying Transactions: The miner verifies the authenticity of each transaction by confirming its input accuracy and ensuring that sufficient funds are available for the transaction.
3) Creating a Block: Once valid transactions are collected, the miner creates a new block. The new block has a timestamp, a nonce, a Markel Root hash representing

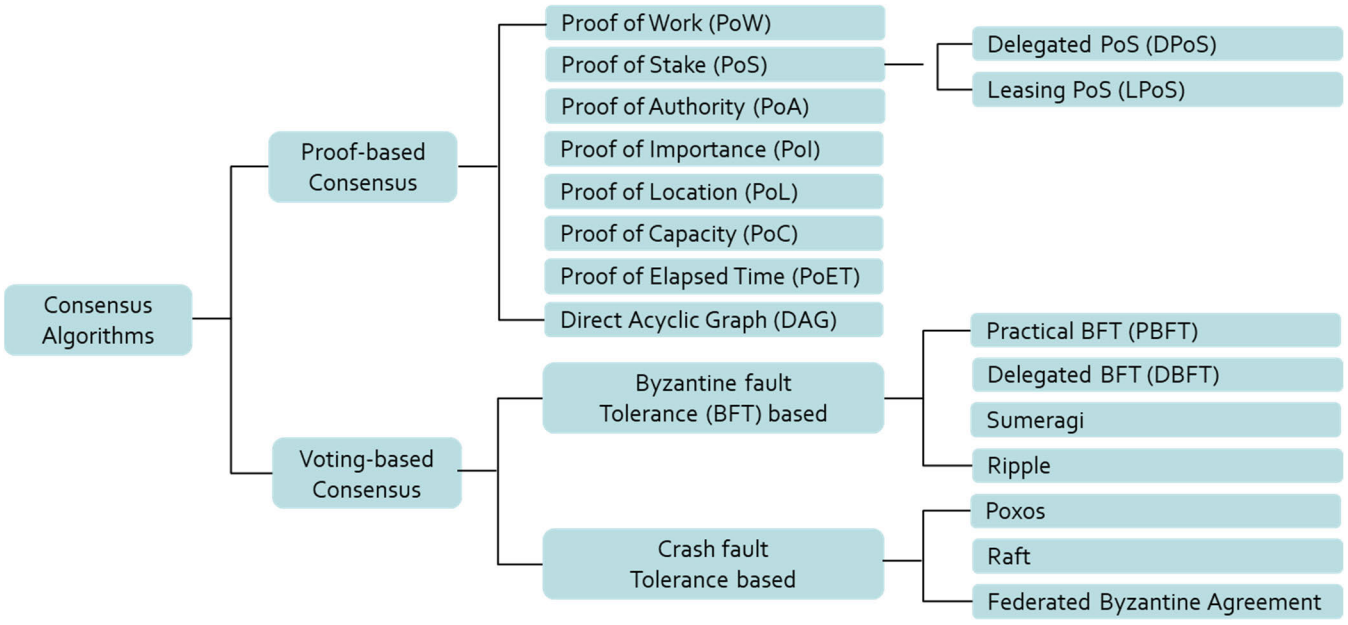



**FIGURE 16.** Classification of popular consensus protocols.

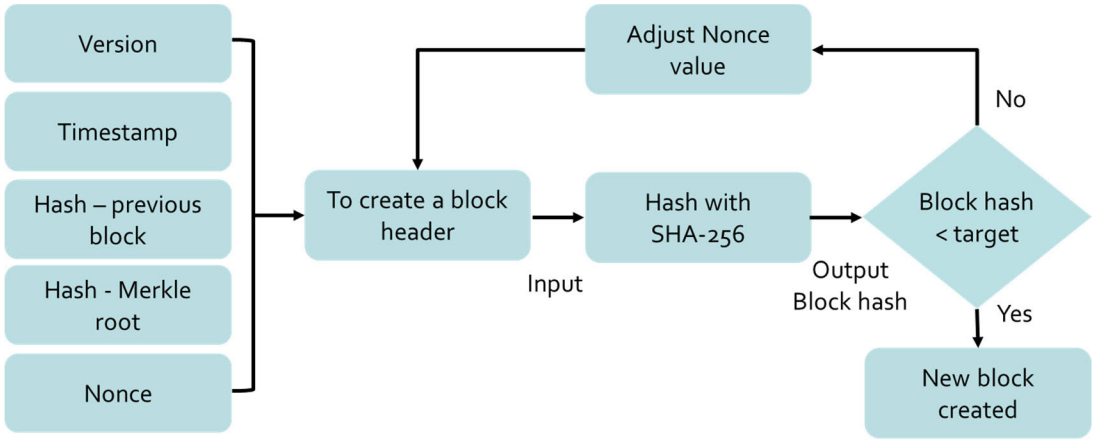



**FIGURE 17.** Proof-of-work (PoW) consensus.

the group of transactions, and the previous block's hash.

4) Finding Nonce Value: Once the block is prepared, the miner discovers the nonce value by producing a hash with the network's predefined difficulty target nBit. The miner achieves this by repeatedly hashing the block header with different nonce values to find the valid hash.
5) Broadcasting the Block: When a valid hash is found, the miner shares the new block with the entire Bitcoin network by broadcasting it. Other nodes in the network validate the block and incorporate it into their records.
6) Receiving the Reward: As compensation for adding a new block to the blockchain, the miner receives a reward in the form of a newly created coin and a transaction fee associated with the transaction processing is also included in the block.

The classification of popular consensus algorithms is presented in Figure 16 below.

### A. PROOF-OF-WORK (POW)

PoW is the most widely adopted consensus algorithm in blockchain, used by Bitcoin and numerous other cryptocurrencies [106]. Miners compete to find the correct nonce value to validate transactions and append blocks to the blockchain. The first miner to solve the puzzle is rewarded with the new cryptocurrency [107]. PoW requires miners to use substantial computational power and energy to solve these puzzles. The

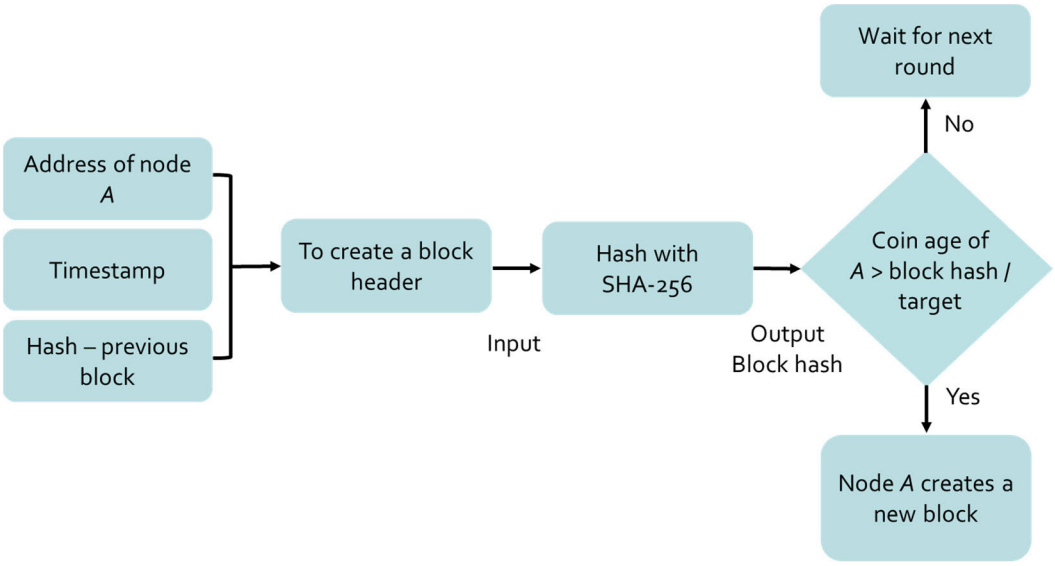



**FIGURE 18.** Proof-of-stake (PoS) consensus.

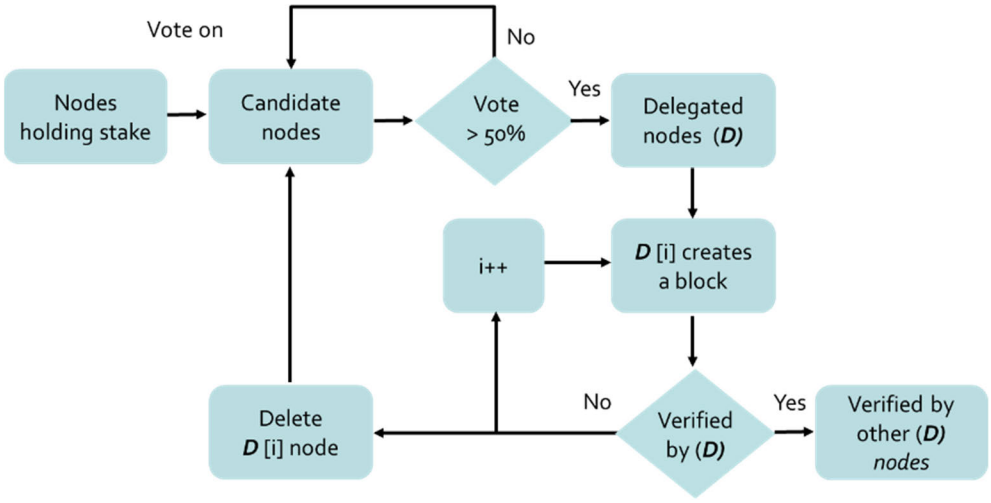



**FIGURE 19.** Delegated proof-of-stake (DPoS) consensus.

miner who completes the puzzle and validates the block is also rewarded [108]. PoW serves as a security mechanism, preventing easy manipulation of the transaction history and ensuring the security of the blockchain network. The difficulty of the puzzles is adjusted to maintain a consistent block creation rate. By ensuring that trustworthy users control most of the network's computational power, PoW makes it challenging for attackers to gain control over the system [109]. PoW incentivizes honest behavior among network users and deters malicious actions by mandating miners to invest resources and compete to solve puzzles. However, PoW is associated with high energy consumption and scalability challenges. As a result, developers and researchers are exploring some alternate consensus algorithms and developing them, such as PoS and others that are more secure, fast, and energy-efficient [110].

Figure 17 illustrates the working mechanism of the PoW consensus protocol.

### B. PROOF-OF-STAKE (POS)

PoS is a well-liked consensus algorithm that many cryptocurrencies use. Based on the amount of cryptocurrency they hold, PoS selects validators to create and validate new blocks. In contrast to PoW, in which miners compete on the basis of computational power, they are willing to stake or lock up [111]. In a Proof-of-Stake consensus, the validators are chosen based on their network stake in order to create new blocks and validate transactions. The chances of being selected as a validator are proportional to the amount of cryptocurrency one holds and is willing to risk [112]. This selection procedure is frequently arbitrary or dependent on several variables, including age and stake size.

Confirming transactions and adding them to the blockchain are the responsibilities of validators. Validators are rewarded with additional cryptocurrency for their participation and network security. This incentive system encourages validators to act honorably and uphold the blockchain's integrity. PoS offers several advantages compared to PoW [110].

Figure 18 depicts the process of the PoS consensus protocol.

One notable benefit is its lower energy consumption since it does not rely heavily on computational power. Additionally, PoS addresses the scalability challenges associated with PoW, allowing for faster transaction processing. However, PoS introduces challenges, such as the nothing at stake problem, where validators may attempt to validate multiple blockchain forks without incurring expenses. Blockchain platforms and cryptocurrencies have implemented different PoS variations, each with unique guidelines and mechanisms. PoS is an alternative consensus algorithm that leverages network participants' stakes and financial incentives to ensure network security and transaction validation [113].

### C. DELEGATED PROOF-OF-STAKE (DPOS)

The DPoS consensus algorithm is used in some blockchain networks as an alternative to PoW and PoS. A more efficient and scalable way to reach consensus is through DPoS, which gives the job of validating blocks to a small group of trusted delegates or nodes. The token holders within the network can vote for delegates responsible for constructing and validating new blocks. Depending on the blockchain protocol, the delegates typically range from a few to a few dozen [114]. During the voting process, token holders can select delegates based on qualities such as reputation, skill, or contributions to the network [115]. Often, the influence or weight of a voter's decision is inversely proportional to the number of tokens they possess. The chosen delegates take turns creating blocks and approving transactions on behalf of the network. By eliminating the requirement for each participant to validate every transaction, DPoS introduces a level of efficiency that allows for higher transaction throughput and reduced network latency as the consensus process becomes faster with a smaller group of trusted delegates. Consequently, DPoS is well-suited for blockchain applications that prioritize scalability and fast transaction processing. In a DPoS consensus, the elected delegates are incentivized to act honorably and prioritize the network's best interests. They may receive compensation for their efforts in generating and validating blocks, which can come from transaction fees or newly created tokens. Token holders retain the power to vote out or replace delegates in subsequent voting rounds if they behave maliciously or fail to fulfill their duties [116].

DPoS has been implemented on various blockchain platforms and cryptocurrencies, each with unique rules and variations. While DPoS offers scalability benefits, it also introduces a level of centralization due to the limited number of delegates involved [110]. Designing and implementing DPoS-based blockchain systems requires careful consideration of how to strike a balance between scalability and decentralization.

Figure 19 illustrates the flow of DPoS consensus.

### D. PROOF-OF-AUTHORITY (POA)

In blockchain networks where the validators or block producers are pre-approved and trusted entities, PoA is a consensus algorithm that is frequently used. To validate and add blocks to the blockchain, PoA relies on the reputation and identity of validators, as opposed to other consensus mechanisms that depend on computational work or stakes. A group of authorized validators, frequently called authorities or signers, are chosen to build new blocks and approve transactions in a PoA system. These authorities are frequently well-known and dependable entities, such as reputable businesses, government organizations, or people [117]. They are chosen based on their standing, experience level, or position within the network. In PoA, the designated authorities propose and validate blocks as part of the block creation process. The ability to create and validate blocks is granted to each authority without the need for intense computation or competition. Authorities decide whether to approve a transaction based on established guidelines and consensus agreements.

The consensus achieved in a PoA system relies on the assumption that trusted authorities will act in the network's best interest [113]. This consensus model offers fast block confirmation times and high transaction throughput since there are no computational or stake-based competition restrictions on the block creation process. One of the key advantages of PoA is its resilience to attacks from malicious actors. The presence of well-known and reliable authorities significantly reduces the likelihood of a 51% or double-spending attack. Concentrating consensus power among a few trusted entities introduces a certain level of centralization. In PoA, validators or authorities may not require traditional mining rewards or transaction fees as incentives. Instead, they are typically motivated by reputational advantages or other strategies. PoA is commonly used in private or consortium blockchains, where network participants are known and have a stake in preserving the security and integrity of the blockchain.

### E. BYZANTINE FAULT TOLERANCE (BFT)

The Byzantine Generals Problem is what BFT is based on. This problem happens when it is hard to reach a consensus because some nodes in a network are acting randomly and inconsistently. In a BFT algorithm, a predetermined set of nodes, often called replicas or validators, participate in the consensus process. These nodes exchange messages to agree on the system's state, such as the transactions' order or the blocks' content in a blockchain. Byzantine fault tolerance means that the nodes can handle and get around the presence of bad or malfunctioning nodes. This is done by using different protocols and mechanisms. These mechanisms typically involve multiple rounds of voting and communication among

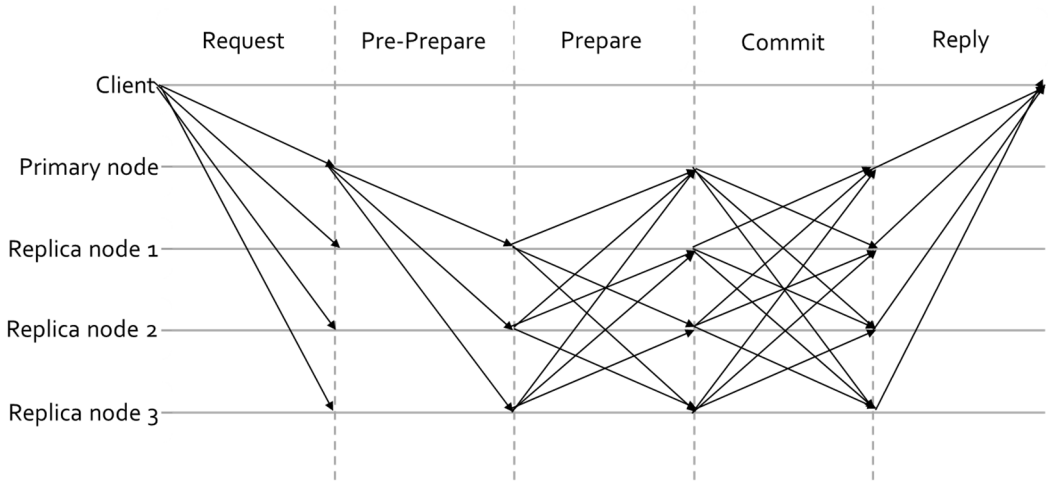



**FIGURE 20.** Practical Byzantine fault tolerance (PBFT) consensus.

the participating nodes [118]. In each round, nodes communicate with each other and share their proposed decisions or values. Based on the received messages and predefined rules, they collectively decide on the agreed-upon value.

The goal is for the non-faulty nodes to agree on the correct value, even when malicious nodes provide contradictory information [119]. BFT algorithms often require a certain number of correctly functioning nodes to reach consensus. For example, a common requirement is that at least two-thirds of the participating nodes behave honestly and consistently. This threshold ensures that most presumed-reliable nodes participate in the consensus decision-making process [120]. BFT provides several benefits in distributed network including blockchain network and some of them are as follows:

1) Fault and Attack Resistance: BFT algorithms make sure that a small number of bad nodes do not change the consensus decision. This defends against attacks and unintentional errors. This resilience strengthens the overall security and reliability of the system.
2) In BFT algorithms, the agreed-upon value is considered final and cannot be changed without the participation of a large number of nodes. This ensures that the state of the system will not change and that it will be safe.
3) Security and Trustworthiness: BFT algorithms make the network safer and more trustworthy by stopping bad nodes or attackers from interfering with the consensus process. This protects the system's integrity and builds trust among network participants in how it works.

BFT algorithms have many different implementations, each with a unique protocol and set of features. Popular BFT algorithms include Tendermint, ByzCoin, and PBFT.

### F. PRACTICAL BYZANTINE FAULT TOLERANCE (PBFT)

The purpose of PBFT is to provide distributed systems with Byzantine fault tolerance. It is often used in permissioned blockchain networks that value scalability, low latency, and quick transaction processing. Traditional Byzantine fault tolerance algorithms have problems like too much communication overhead and slow performance. PBFT resolves these problems by using a more efficient and useful consensus approach [45]. In PBFT, the network comprises multiple nodes known as replicas or validators, which participate in the consensus process [118]. Applications that value speed, scalability, and fault tolerance in permissioned blockchain networks with known and reliable participants are well-suited for PBFT [110].

Figure 20 depicts the practical byzantine fault tolerance consensus.

In PBFT, consensus is achieved through a series of rounds, each involving the following steps:

1) Request: The consensus process begins when a client initiates a request by sending a transaction proposal to the replicas.
2) Pre-Prepare: Upon receiving the client's request, the primary replica disseminates a pre-prepare message to the other replicas. This message contains the proposed request and a sequence number assigned by the primary replica.
3) Prepare: Replicas validate the request and broadcast prepared messages to express their agreement with the proposed request in response to the pre-prepare message.
4) Commit: Upon receiving a sufficient number of prepared messages, a replica broadcasts a commit message to indicate its readiness to finalize the request.
5) Reply: Once a replica receives commit messages from the required number of replicas, it responds to the client, who considers the request fulfilled.

The key features and benefits of PBFT include:

1) Enhanced Performance: PBFT exhibits superior performance in terms of reduced latency and increased throughput compared to alternative consensus

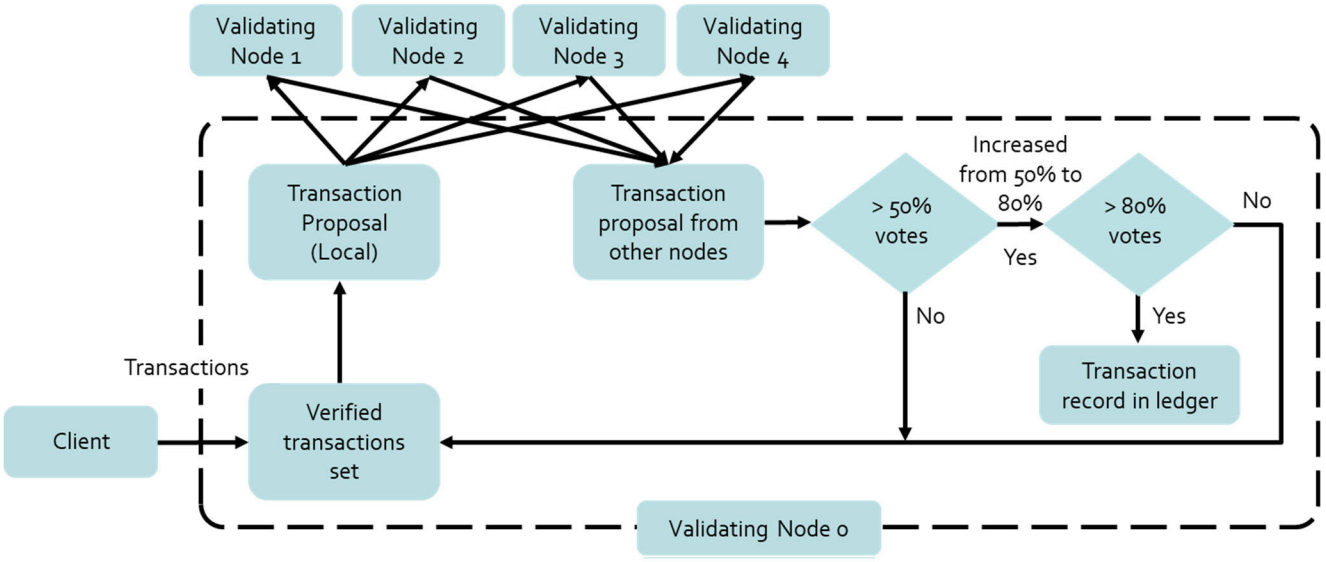



**FIGURE 21.** Ripple protocol consensus algorithm (RPCA) consensus.

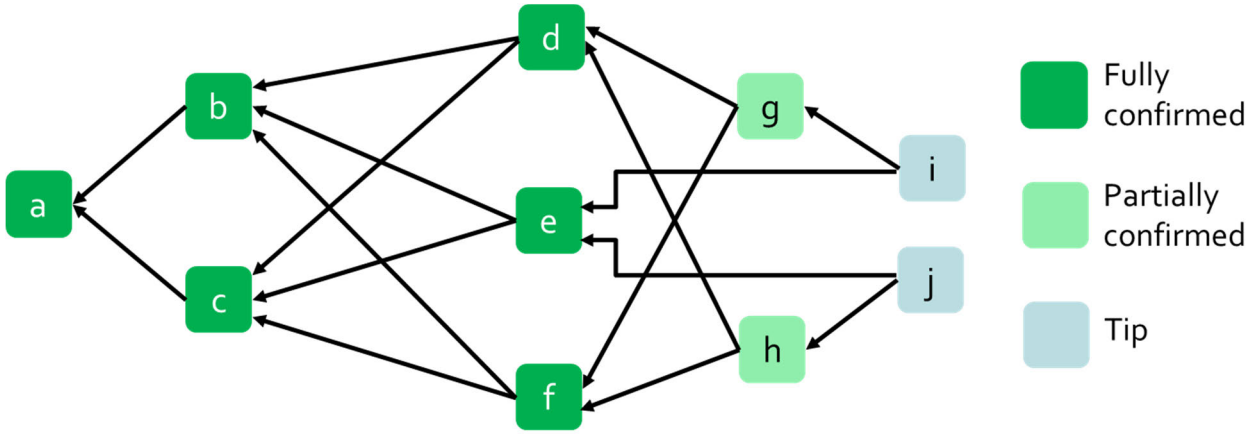



**FIGURE 22.** Tangle consensus.

**TABLE 6.** Comparative analysis of tangle and Bitcoin (PoW) consensus protocols.

| | Tangle | Bitcoin (PoW) |
|---|---|---|
| Transaction confirmation Speed | Fast | Slow |
| Transaction Fee | No Fee | High |
| Micro Transaction | Suitable for micro-transactions | Highly expensive |
| Miners' role | N/A | Essential |
| Information storage | Nodes | Blocks |
| Fork | Not supported | Supported |
| Distributed consensus | On payment | With the help of minors |
| Scalable | Yes | No |
| Offline capability | Yes | No |
| Transaction confirmation time | Instant | With lower fees, more confirmation time, and vice versa |

algorithms. It is done through a message-based protocol and concurrent processing of requests.

2) Improved Scalability: PBFT can scale to accommodate many replicas while maintaining efficient transaction processing, making it well-suited for highly distributed systems.
3) Byzantine Fault Tolerance: PBFT ensures reaching consensus even if malicious or faulty replicas exist.

**TABLE 7. Comparison of different consensus protocols [39].**

| Method | Accessibility | Decenter-lization | Scalability | Fault Tolerance | Throughput | Latency | Computing Overhead | Network Overhead | Storage Overhead |
|---|---|---|---|---|---|---|---|---|---|
| PoW | Public, PL | H | H | <25% | L | H | H | L | H |
| PoS | Public, P/PL | H | H | < 50% | L | M | M | L | H |
| DPoS | Public, PL | M | H | <51% | H | M | M | N/A | H |
| LPoS | Public, PL | H | H | <51% | L | M | M | L | H |
| PoA | Public, PL | H | H | <51% | L | M | H | L | H |
| PoI | Public, PL | H | H | <51% | H | M | L | L | H |
| PoC | Public, PL | H | H | N/A | L | H | L | L | VH |
| PoET | Private, P/PL | M | H | N/A | H | L | L | L | H |
| Tangle | Public, PL | M | H | <33% | H | L | L | N/A | H |
| PBFT | Private, P | M | L | <33% | H | L | L | H | H |
| DBFT | Private, P | M | H | <33% | H | M | L | H | H |
| Ripple | Public, PL | H | H | <20% | H | M | L | M | H |
| Raft | Private, P | M | H | <50% | H | L | L | N/A | H |
| Stellar | Pubic, PL | H | H | <20% | H | M | L | M | H |
| PoB | Public, PL | H | H | <25% | L | H | M | L | H |

P = Permission, PL = Permissionless, VH = Very high, H = High, M = Medium, L = Low

It guarantees that all correct replicas agree on the same set of requests.

4) Irreversible Finality: Once a request is committed in PBFT, it is considered finalized and cannot be reversed without a significant portion of the replicas failing simultaneously.
5) Resource Efficiency: PBFT requires fewer computational resources for activities such as mining and proof-of-work calculations compared to certain other consensus algorithms. This results in improved energy efficiency [121].

### G. RIPPLE PROTOCOL CONSENSUS ALGORITHM (RPCA)

Another well-known consensus protocol that RIPPLE created is RPCA. Ripple uses the RPCA consensus mechanism to facilitate the agreement and validation of transactions within distributed networks [122]. RPCA also addresses many issues facing current systems to facilitate cross-border payment processes. RPCA streamlines the agreement and validation of transactions within the Ripple network, enabling secure, efficient, and decentralized transaction processing, which facilitates fast and reliable cross-border payments and other financial transactions [123].

Figure 21 illustrates the RPCA consensus.

The underlying principles of RPCA's consensus are [122]:

1) Validator Nodes: The Ripple network consists of validator nodes that determine the validity and order of transactions. Users of the network choose these nodes, run by individuals, organizations, or financial institutions.
2) Unique Node List (UNL): Validators maintain their own UNL, a subset of trusted validators considered during consensus. UNL ensures communication with reliable nodes.
3) Iterative Process: RPCA operates in rounds, with validators engaging in iterative steps to propose, validate, and agree on the transaction set for the ledger.
4) Agreement Process: Validators exchange messages and adjust candidate transactions to converge on a majority decision. The aim is to reach a consensus on a single valid transaction set.
5) Transaction Ordering: RPCA allows validators to order transactions based on rules and priorities independently. This ordering ensures consistency and resolves conflicts.
6) Fault Tolerance: RPCA is Byzantine fault-tolerant, accommodating malicious or faulty nodes. Consensus ensures agreement on the ledger state despite Byzantine failures.
7) Ripple-Validated Ledger: Once the consensus is reached, the validated transaction set is added to the Ripple-validated ledger, which provides a shared and immutable record.

### H. TANGLE

IOTA utilizes a distinctive consensus process known as tangle consensus instead of relying on traditional methods like PoW or PoS commonly found in blockchain technology [60]. The foundation of IOTA's consensus is the Tangle, a Directed Acyclic Graph (DAG) structure [124]. Transactions in IOTA are interconnected, forming a network of transactions resembling a web. IOTA employs a tangle-based consensus mechanism called tip selection instead of relying on miners to establish consensus. When a new transaction is initiated, participants in the network validate the authenticity of two previous transactions, thereby confirming their legitimacy. This process allows for high scalability and eliminates the need for transaction fees [104]. The Tangle consensus mechanism prioritizes decentralization, network security, and

scalability. It is known as the weighted cumulative weight algorithm. As the number of transactions grows, the Tangle aims to enhance its security and efficiency. Each new transaction indirectly validates earlier transactions, bolstering the network's overall security. It is important to note that the Tangle consensus mechanism IOTA employs is unique to the network and differs from the PoW or PoS consensus systems used by other cryptocurrencies such as Bitcoin or Ethereum [125]. The consensus process in Tangle is completed in the following steps [126]:

1) Transaction Validation: When a participant intends to conduct a transaction on the IOTA network, they must validate two preceding transactions. This validation procedure involves confirming the authenticity of the transactions by checking their digital signatures and ensuring they do not clash with any existing transactions.
2) Tip Selection: As part of the transaction process, participants are required to select and validate two tips, which are unconfirmed transactions. Generally, tips are chosen based on specific criteria, such as the transaction's weight (cumulative difficulty).
3) Adding Transaction: Once the tips are chosen, the participant generates a new transaction that refers to them. This newly created transaction is included in the DAG and becomes an integral part of the Tangle structure.
4) Proof of Work: Before broadcasting the new transaction to the network, the participant must perform a brief Proof of Work task. This step aims to safeguard against spam attacks by necessitating a specific level of computational work to add each transaction to Tangle.
5) Confirmation and Approval: As more participants generate transactions, they will refer to and indirectly endorse previously created transactions. Through these references to past transactions, the newly created transaction contributes to their cumulative weight, thereby reinforcing their security and confirmation.
6) Network Consensus: Over time, as additional transactions are incorporated into the Tangle, the consensus is achieved by considering the overall weight of the transactions. Transactions with higher cumulative weights are regarded as more secure and confirmed. Participants can evaluate the level of trust in a transaction by assessing the number of validations and approvals obtained from other transactions within the Tangle.

The deeper a transaction is within the Tangle, or the more transactions are intertwined, the more finalized and secure it becomes [126]. While the blockchain does not allow for two conflicting transactions, the Tangle may temporarily contain such transactions. These conflicting transactions can exist in the Tangle for a short time until nodes with integrity detect the conflict. Once a conflict is identified, nodes must decide which transaction to process [128]. For the Tangle to function as a true consensus protocol, it is necessary to resolve these conflicting transactions. In Figure 22, the transaction represented by the darker green color is more deeply embedded in the Tangle and is expected to eventually receive approval from all nodes. The decision regarding which transaction to accept should be determined using a consensus protocol [126]. In Figure 22, the light green transactions validate the dark green transactions, illustrating this concept. Table 6 presents a comparison between the Tangle and Bitcoin consensus protocols.

Table 7 summarizes the features comparison of popular consensus protocols.

## VII. DISCUSSION

The unique features of blockchain technology make it applicable to various domains such as IoT, healthcare, data storage, inventory tracking, finance, etc. Adapting blockchain to specific application requirements presents a challenge, requiring new or customized implementations. In this research work, we have identified areas where specific research challenges need to be addressed, including scalability issues, data privacy protection [129], increasing throughput, reducing latency, overhead computing, and network and storage limitations. As blockchain is considered a significant invention in the ICT community, studying the underlying cryptographic primitives in the blockchain is crucial for understanding its security and privacy [130]. By examining various studies, cryptographic primitives have been classified into three main categories, with an evaluation of their functionality, usage, and evolution.

Despite the advancements in blockchain and its applications, there are still unresolved challenges related to security and privacy [131]. These issues encompass the need for lightweight algorithms that can operate effectively, finding improved methods to counteract the unexpected accumulation of computational power, and tackling the paradoxical situation where cryptographic algorithms have a limited lifespan and, at the same time, blockchain claims to provide everlasting and tamper-proof characteristics [132]. These challenges require urgent attention from researchers and development companies. While blockchain offers several advantages, such as decentralization, transparency, and immutability, it also has limitations and challenges. These limitations and challenges are important to address as they are directly and indirectly related to blockchain's capacity for security and privacy. Some direct limitations and challenges are listed as follows:

1) Privacy Concerns: While blockchain offers transparency, the public nature of transactions can compromise participants' privacy. Sensitive information recorded on the blockchain may be accessible to anyone, potentially leading to privacy concerns.
2) Security Vulnerabilities: Although blockchain networks are designed to be secure, vulnerabilities can still exist. Smart contract bugs, 51% attacks, and vulnerabilities in specific implementations can pose risks to the security of the blockchain.

3) High Upfront Cost: Ensuring blockchain security depends on some specialized hardware and software that require a huge upfront cost.

Indirect limitations and challenges could have resulted in structural changes in the blockchain. Therefore, any flaw or weakness in the blockchain network could come to the surface. Addressing them simultaneously would not be possible, and there are chances some of them may not be noticed, leading to an open end for attacks. Some indirect limitations of and challenges to blockchain are as follows:

1) Scalability: As the volume of transactions and participants in blockchain networks, particularly public networks like Bitcoin and Ethereum, grows, scalability concerns may arise. These issues can manifest as longer transaction processing times and increased fees.
2) Energy Consumption: PoW consensus algorithms employed in certain blockchains demand substantial computational power, resulting in significant energy consumption. Consequently, concerns have been raised regarding the environmental consequences associated with the energy usage of blockchain technology.
3) Governance and Regulation: The decentralized nature of blockchain can make governance and regulation challenging. Issues related to legal compliance, jurisdiction, and resolving disputes within blockchain networks require careful consideration.
4) Adoption and Usability: Blockchain technology is still relatively new and complex, hindering widespread adoption. User interfaces and tools for interacting with blockchain systems must be more user-friendly to enhance usability.

### A. TECHNICAL CHALLENGES

Although blockchain assists in ensuring confidentiality and integrity (e.g., resistance to unauthorized modifications) of data, there are several security issues and challenges that arise and present threats to it. In the blockchain, cryptographic primitives are the essential building blocks for ensuring data confidentiality and integrity. However, these primitives are exposed to different challenges as technology advances. Some of the current challenges and threats include the followings:

1) Quantum Threats: Many blockchain protocols rely on cryptographic algorithms such as RSA and ECC to ensure data security and privacy. These primitives mainly rely on integer factorization and discrete logarithms, and quantum computing nowadays can factor large numbers exponentially faster than classical computers [133]. This could indeed compromise the security requirements, e.g., confidentiality and integrity, of stored data on the blockchain. For long-term safety and to protect against these kinds of threats, experts and researchers are now focusing on making and using cryptographic primitives that are resistant to quantum attacks, like code-based and lattice-based cryptography [134].
2) Side-Channel Attacks (SCA): SCA is quite easy to launch, and it uses less power. It is an important challenge to the blockchain and is mainly used in IoT during the exchange of data [135]. Attackers can extract relevant information during cryptographic operations through different attacks, such as power consumption attacks, electromagnetic radiation attacks, timing variation attacks, acoustic cryptanalysis attacks, and differential fault analysis [136]. For example, a simple power consumption analysis may expose cryptographic keys to malicious parties [137], and continuous SCA can easily break any unprotected cryptographic implementations [138].
3) Sustainability: Cryptographic primitives need to withstand attacks over time despite advances in computational power, e.g., quantum computing. Modern cryptography follows Kerckhoff's principle, which goes as follows: a cryptographic system should be secure even if everything about the system, except the key, is public knowledge. Hence, these primitives should be continuously cryptanalyzed to withstand any attempt to exploit new vulnerabilities [139].
4) Key Management: It is one of the urging issues in cryptographic primitives. The blockchain uses two keys: a public key and a private key. However, if the private key is lost, there is no way to retrieve it due to the decentralization of the blockchain [140]. Proper key management will reduce the threat of private key theft, but storing these keys in third-party services may increase the risk due to the possibility of exploiting vulnerabilities in these services [141].
5) Implementation Vulnerabilities: In response to quantum threats, it is very important to ensure that implemented quantum cryptographic algorithms are secure and effective to withstand emerging attacks. In [142], the authors studied the implementations of different types of quantum cryptographic primitives, and their study revealed security vulnerabilities that an attacker could exploit.
6) Interoperability: The expansion of the blockchain ecosystem requires secure communication and interoperability between different blockchain networks. For instance, all blockchains must support compatible implementations of the necessary cryptographic primitives for digital signatures to be valid. If they do not, this could cause problems when signing and verifying those signatures [143].

### B. BROKEN OR COMPROMISED CRYPTOGRAPHIC PRIMITIVES AND CONTINGENCIES

Cryptocurrency networks rely heavily on cryptographic primitives to function, and these primitives do not last forever and become weaker with time, making them vulnerable to attacks [76]. This is due to advances in hardware with high computational powers and the most recent advancements in hacking

techniques. Developers must remain vigilant to anticipate such breakdowns. As a result, it is prudent to anticipate that the cryptographic primitives used by Bitcoin will be partially, if not completely, broken over time [73]. In anticipation of such scenarios, the Bitcoin community has issued guidelines and contingency plans [144].

#### 1) HASH FUNCTION

The hash function has a 256-bit output and requires SHA256 to be applied as $SHA256(SHA256(x))$. It is used for Bitcoin mining as well as hashing transactions and Merkle Tree. The second hash function is used in the scripts with the combination of RIPEMD160 and SHA256 as RIPEMD160(SHA256(x)), and it has a 160-bit output. [145]. If the hash function is broken or compromised, only a few days of effort are required to break or bypass OP_CHECKSIG, which is used for hashing the transactions before signing. Attackers can also split the network by generating identical transactions or blocks with the same hashes. They may also generate new blocks extremely quickly.

If the hash function in Bitcoin is broken or compromised, the users will be instructed to close their applications until further instructions are given. The OP_CHECKSIG function will be modified to use different hash schemes on the old blocks [144]. The known public key in the old chain with at least one unspent output will be hardcoded, and when a transaction from the new chain spends one of those old chains, the hash will be replaced by the hardcoded public key. The Markel root will also be hashed into the new chain. A new hashing scheme will be implemented for all of its hashing requirements [144].

#### 2) DIGITAL SIGNATURES

Bitcoin signs transactions with the main hash using the ECDSA digital signature scheme with secp256k1 parameters [73]. A signature of HM(m) must also be valid for HM(m′ ) to break the integrity of the signature scheme. If a signature is verified using a different key, the address hashes of the two public keys must be the same [145]. Attackers may be able to sign the public key with only a few days of effort.

In many cases, attackers can spend money that does not belong to them. Transactions made to previously unknown addresses may be protected if SHA-256 and RIPEMD-160 are not compromised. After only a few days of work, attackers can use their private key to sign for a public key that they do not own [144]. If attackers cannot easily obtain the private key from the public key and a stronger algorithm capable of using ECDSA keys is available, switching to a stronger algorithm is an option. When the new version of Bitcoin is launched, it should automatically send all previous transactions to a new location using the new algorithm.

## VIII. CONCLUSION

This study reviews the cryptographic building blocks that underpin both blockchain technology and Bitcoin. These cryptographic primitives are at the core of the blockchain, which helps to ensure that the technology is reliable and trustworthy. The blockchain's security barriers are robust enough due to the combination of SHA-256, RIPEMD160, and ECDSA, which together ensure the security of the blockchain. As far as the current state of the security of blockchain is concerned, it is still flawless, and there is no evidence that, at any stage, the security of blockchain has been breached or compromised on a large scale. In the past, it was reported that 51% of miners pooled themselves to gain more control over the mining process. The so-called "Man in the Middle" attack is similar to the others and has been documented numerous times; however, such attacks can be avoided with the help of strong network security. The data associated with Bitcoin is secure and protected by cryptographic primitives, and it is still robust enough to withstand any method of attack that is currently known to hackers. On the whole, we are confident that blockchain technology and Bitcoin are quite secure because they have implemented the most effective combination of security algorithms and cryptography techniques that are both impossible or nearly infeasible to hack or reverse.

Ransomware is an emerging threat to digital devices and blockchain can also play a vital role in mitigating ransomware attacks to protect digital devices, networks and information [146]. Further, quantum computing should be taken into consideration as well. This should emphasize the focus on investigating the vulnerabilities and shortcomings in the cryptographic primitives and consensus algorithms that are used to protect the blockchain. It has been suggested that the security levels of the blockchain and cryptographic primitives be re-examined with the processing speed of quantum computing in mind. Quantum computing could represent either a potential threat or a potential opportunity for the future of the blockchain, which has been relatively secure for now. During the time of transition, it is also recommended that a contingency plan is developed, implemented and tested to be ready in the event of a disruption.

## REFERENCES

[1] M. I. Sarwar, K. Nisar, I. Khan, and D. Shehzad, "Blockchains and triple-entry accounting for B2B business models," *Ledger*, vol. 8, pp. 37–57, Jul. 2023, doi: 10.5195/ledger.2023.288.

[2] F. J. de Haro-Olmo, Á. J. Varela-Vaca, and J. A. Álvarez-Bermejo, "Blockchain from the perspective of privacy and anonymisation: A systematic literature review," *Sensors*, vol. 20, no. 24, pp. 1–21, Dec. 2020, doi: 10.3390/s20247171.

[3] M. I. Sarwar, K. Nisar, and A. Khan, "Blockchain—From cryptocurrency to vertical industries—A deep shift," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Dalian, China, Sep. 2019, pp. 1–4, doi: 10.1109/ICSPCC46631.2019.8960795.

[4] P. Ratta, A. Kaur, S. Sharma, M. Shabaz, and G. Dhiman, "Application of blockchain and Internet of Things in healthcare and medical sector: Applications, challenges, and future perspectives," *J. Food Qual.*, vol. 2021, pp. 1–20, May 2021, doi: 10.1155/2021/7608296.

[5] M. N. M. Bhutta, A. A. Khwaja, A. Nadeem, H. F. Ahmad, M. K. Khan, M. A. Hanif, H. Song, M. Alshamari, and Y. Cao, "A survey on blockchain technology: Evolution, architecture and security," *IEEE Access*, vol. 9, pp. 61048–61073, 2021, doi: 10.1109/ACCESS.2021.3072849.

[6] R. Sheldon. (Aug. 2021). *A Timeline and History of Blockchain Technology*. Accessed: Oct. 2, 2021. [Online]. Available: https://www.techtarget.com/whatis/feature/A-timeline-and-history-of-blockchain-technology

[7] B. S. Tan and K. Y. Low, "Blockchain as the database engine in the accounting system," *Austral. Accounting Rev.*, vol. 29, no. 2, pp. 312–318, Jun. 2019, doi: 10.1111/auar.12278.

[8] D. Tait, T. Lynch, R. W. J. Gonzalez-Medina, D. Green, E. M. Basslim, and H. Shah, "Blockchain vertical opportunities report—2018," S&P Global (Formerly IHS Markit). Accessed: May 30, 2022. [Online]. Available: https://cdn.ihs.com/www/pdf/1118/abstract-blockchain-vertical-opportunities-report-2018.pdf

[9] I.-C. Lin and T.-C. Liao, "A survey of blockchain security issues and challenges," *Int. J. Netw. Secur.*, vol. 19, no. 5, pp. 653–659, Sep. 2017, doi: 10.6633/IJNS.201709.19(5).01.

[10] I. Acharjamayum, R. Patgiri, and D. Devi, "Blockchain: A tale of peer to peer security," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Bengaluru, India, Nov. 2018, pp. 609–617, doi: 10.1109/SSCI.2018.8628826.

[11] K. Rabah, "Convergence of AI, IoT, big data and blockchain: A review," *lake Inst. J.*, vol. 1, no. 1, pp. 1–18, 2018.

[12] O. Ali, A. Jaradat, A. Kulakli, and A. Abuhalimeh, "A comparative study: Blockchain technology utilization benefits, challenges and functionalities," *IEEE Access*, vol. 9, pp. 12730–12749, 2021, doi: 10.1109/ACCESS.2021.3050241.

[13] S. Johar, N. Ahmad, W. Asher, H. Cruickshank, and A. Durrani, "Research and applied perspective to blockchain technology: A comprehensive survey," *Appl. Sci.*, vol. 11, no. 14, p. 6252, Jul. 2021, doi: 10.3390/app11146252.

[14] D. L. Chaum, "Computer systems–established, maintained, and trusted by mutually suspicious groups," Ph.D. thesis, Graduate Division, Univ. California, Berkeley, CA, USA, 1982.

[15] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," *J. Cryptol.*, vol. 3, no. 2, pp. 99–111, Jan. 1991, doi: 10.1007/bf00196791.

[16] D. Bayer, S. Haber, and W. S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," in *Sequences II*. New York, NY, USA: Springer, 1993, pp. 329–334, doi: 10.1007/978-1-4613-9323-8_24.

[17] R. Sharma. *Bit Gold*. Accessed: Dec. 25, 2021. [Online]. Available: https://www.investopedia.com/terms/b/bit-gold.asp

[18] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: Aug. 4, 2018. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[19] M. I. Sarwar, K. Nisar, S. Andleeb, and M. Noman, "Blockchain—A crypto-intensive technology—A review," in *Proc. 35th Int. Bus. Inf. Manage. Assoc. (IBIMA) Conf.*, Seville, Spain, 2020, pp. 14803–14809.

[20] V. Buterin. *Ethereum Whitepaper*. Accessed: Jun. 29, 2021. [Online]. Available: https://ethereum.org/en/whitepaper

[21] IBM.com. *What is Hyperledger Fabric?* Accessed: Aug. 19, 2020. [Online]. Available: https://www.ibm.com/topics/hyperledger

[22] C. Laroiya, D. Saxena, and C. Komalavalli, *Applications of Blockchain Technology*. New Delhi, India: INC, 2020, doi: 10.1016/B978-0-12-819816-2.00009-5.

[23] Z. Liu, "Literature review of supply chain finance based on blockchain perspective," *Open J. Bus. Manage.*, vol. 9, no. 1, pp. 419–429, 2021, doi: 10.4236/ojbm.2021.91022.

[24] D. Levis, F. Fontana, and E. Ughetto, "A look into the future of blockchain technology," *PLoS ONE*, vol. 16, no. 11, Nov. 2021, Art. no. e0258995, doi: 10.1371/journal.pone.0258995.

[25] M. I. Sarwar, M. W. Iqbal, T. Alyas, A. Namoun, A. Alrehaili, A. Tufail, and N. Tabassum, "Data vaults for blockchain-empowered accounting information systems," *IEEE Access*, vol. 9, pp. 117306–117324, 2021, doi: 10.1109/ACCESS.2021.3107484.

[26] J. Kolb, M. AbdelBaky, R. H. Katz, and D. E. Culler, "Core concepts, challenges, and future directions in blockchain: A centralized tutorial," *ACM Comput. Surv.*, vol. 53, no. 1, pp. 1–39, Jan. 2021, doi: 10.1145/3366370.

[27] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018, doi: 10.1504/ijwgs.2018.095647.

[28] M. Attaran, "Blockchain technology in healthcare: Challenges and opportunities," *Int. J. Healthcare Manage.*, vol. 15, no. 1, pp. 70–83, Jan. 2022, doi: 10.1080/20479700.2020.1843887.

[29] P. Dutta, T.-M. Choi, S. Somani, and R. Butala, "Blockchain technology in supply chain operations: Applications, challenges and research opportunities," *Transp. Res. E, Logistics Transp. Rev.*, vol. 142, Oct. 2020, Art. no. 102067, doi: 10.1016/j.tre.2020.102067.

[30] 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.*, Dec. 2018, pp. 17–21, doi: 10.1145/3301403.3301407.

[31] D. Dave, S. Parikh, R. Patel, and N. Doshi, "A survey on blockchain technology and its proposed solutions," *Proc. Comput. Sci.*, vol. 160, pp. 740–745, Jan. 2019, doi: 10.1016/j.procs.2019.11.017.

[32] M. U. Javed, M. Rehman, N. Javaid, A. Aldegheishem, N. Alrajeh, and M. Tahir, "Blockchain-based secure data storage for distributed vehicular networks," *Appl. Sci.*, vol. 10, no. 6, pp. 1–22, Mar. 2020, doi: 10.3390/app10062011.

[33] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When intrusion detection meets blockchain technology: A review," *IEEE Access*, vol. 6, pp. 10179–10188, 2018, doi: 10.1109/ACCESS.2018.2799854.

[34] A. Prashanth Joshi, M. Han, and Y. Wang, "A survey on security and privacy issues of blockchain technology," *Math. Found. Comput.*, vol. 1, no. 2, pp. 121–147, 2018, doi: 10.3934/mfc.2018007.

[35] A. I. Baba, S. Neupane, F. Wu, and F. F. Yaroh, "Blockchain in accounting: Challenges and future prospects," *Int. J. Blockchains Cryptocurrencies*, vol. 2, no. 1, pp. 44–67, 2021, doi: 10.1504/ijbc.2021.117810.

[36] H. An, "Design and analysis of decentralized public key infrastructure with quantum-resistant signatures," M.S. thesis, Korea Adv. Inst. Sci. Technol., Daejeon, South Korea, 2018.

[37] U. S. S. Gallersdörfer, "Analysis of use cases of blockchain technology in legal transactions," M.S. thesis, Dept. Inform., Tech. Univ. Munich, Munich, Germany, 2017.

[38] A. Jeppsson and O. Olsson, "Blockchains as a solution for traceability and transparency," M.S. thesis, Dept. Des. Sci., Lund Univ., Lund, Sweden, 2017.

[39] S. Zhang and J.-H. Lee, "Analysis of the main consensus protocols of blockchain," *ICT Exp.*, vol. 6, no. 2, pp. 93–97, Jun. 2020, doi: 10.1016/j.icte.2019.08.001.

[40] L. Wang, X. Shen, J. Li, J. Shao, and Y. Yang, "Cryptographic primitives in blockchains," *J. Netw. Comput. Appl.*, vol. 127, pp. 43–58, Feb. 2019, doi: 10.1016/j.jnca.2018.11.003.

[41] H. Guo and X. Yu, "A survey on blockchain technology and its security," *Blockchain, Res. Appl.*, vol. 3, no. 2, Jun. 2022, Art. no. 100067, doi: 10.1016/j.bcra.2022.100067.

[42] S. Rahmadika and K.-H. Rhee, "Blockchain technology for providing an architecture model of decentralized personal health information," *Int. J. Eng. Bus. Manage.*, vol. 10, Jan. 2018, Art. no. 184797901879058, doi: 10.1177/1847979018790589.

[43] K. Korpela, J. Hallikas, and T. Dahlberg, "Digital supply chain transformation toward blockchain integration," in *Proc. 50th Hawaii Int. Conf. Syst. Sci.*, Honolulu, HI, USA, Jan. 2017, pp. 4182–4191, doi: 10.24251/HICSS.2017.506.

[44] V. Buterin, "On public and private blockchains," Etherum Foundation Blog, Tech. Rep., 2015. Accessed: Mar. 14, 2022. [Online]. Available: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains

[45] J. Yusoff, Z. Mohamad, and M. Anuar, "A review: Consensus algorithms on blockchain," *J. Comput. Commun.*, vol. 10, no. 9, pp. 37–50, 2022, doi: 10.4236/jcc.2022.109003.

[46] M. Daghmehchi Firoozjaei, A. Ghorbani, H. Kim, and J. Song, "Hybridge: A hybrid blockchain for privacy-preserving and trustful energy transactions in Internet-of-Things platforms," *Sensors*, vol. 20, no. 3, pp. 1–25, Feb. 2020, doi: 10.3390/s20030928.

[47] H. Xiong, T. Dalhaus, P. Wang, and J. Huang, "Blockchain technology for agriculture: Applications and rationale," *Frontiers Blockchain*, vol. 3, pp. 1–7, Feb. 2020, doi: 10.3389/fbloc.2020.00007.

[48] S. S. Sabry, N. M. Kaittan, and I. Majeed, "The road to the blockchain technology: Concept and types," *Periodicals Eng. Natural Sci. (PEN)*, vol. 7, no. 4, pp. 1821–1832, Dec. 2019, doi: 10.21533/pen.v7i4.935.

[49] D. Aggarwal, G. Brennen, T. Lee, M. Santha, and M. Tomamichel, "Quantum attacks on bitcoin, and how to protect against them," *Ledger*, vol. 3, pp. 1–21, Oct. 2018, doi: 10.5195/ledger.2018.127.

[50] D. Bryson, D. Penny, D. C. Goldenberg, and G. Serrao, "Blockchain technology for government," MITRE Corp., McLean, VA, USA, MITRE Tech. Rep. MTR1800046, 2017. Accessed: Apr. 1, 2022. [Online]. Available: https://apps.dtic.mil/sti/trecms/pdf/AD1108067.pdf

[51] I. Abu-Elezz, A. Hassan, A. Nazeemudeen, M. Househ, and A. Abd-Alrazaq, "The benefits and threats of blockchain technology in healthcare: A scoping review," *Int. J. Med. Informat.*, vol. 142, Oct. 2020, Art. no. 104246, doi: 10.1016/j.ijmedinf.2020.104246.

[52] S. Demirkan, I. Demirkan, and A. McKee, "Blockchain technology in the future of business cyber security and accounting," *J. Manage. Analytics*, vol. 7, no. 2, pp. 189–208, Apr. 2020, doi: 10.1080/23270012.2020.1731721.

[53] GitHub. *Ark*. Accessed: Nov. 22, 2022. [Online]. Available: https://github.com/ArkEcosystemArchive/ark-node

[54] *Binance Coin*. Accessed: Dec. 12, 2022. [Online]. Available: https://github.com/topics/binance-coin

[55] *Bitcoin Cash*. Accessed: Jan. 2, 2023. [Online]. Available: https://github.com/topics/bitcoin-cash

[56] *Cardano Foundation*. Accessed: Jan. 2, 2023. [Online]. Available: https://github.com/cardano-foundation

[57] E. Duffield and D. Diaz. *Dash: A Privacy-Centric Crypto-Currency*. Accessed: Nov. 5, 2022. [Online]. Available: https://www.exodus.com/assets/docs/dash-whitepaper.pdf

[58] *Dogecoin*. Accessed: Nov. 9, 2022. [Online]. Available: https://github.com/dogecoin/dogecoin

[59] *Electroneum*. Accessed: Nov. 6, 2022. [Online]. Available: https://electroneum.com

[60] *IOTA*. Accessed: Nov. 5, 2022. [Online]. Available: https://github.com/iotaledger

[61] *Komodo*. Accessed: Nov. 1, 2022. [Online]. Available: https://github.com/SuperNETorg/komodo

[62] *Litecoin*. Accessed: Nov. 6, 2022. [Online]. Available: https://litecoin.org

[63] *Ripple*. Accessed: Oct. 29, 2022. [Online]. Available: https://github.com/ripple/

[64] *Solana Labs*. Accessed: Jan. 1, 2023. [Online]. Available: https://github.com/solana-labs

[65] *Stellar*. Accessed: Dec. 30, 2022. [Online]. Available: https://github.com/stellar

[66] *Tether*. Accessed: Jan. 2, 2023. [Online]. Available: https://github.com/topics/tether

[67] *Zcash*. Accessed: Nov. 5, 2022. [Online]. Available: https://github.com/zcash

[68] *Zcoin*. Accessed: Dec. 3, 2022. [Online]. Available: https://github.com/topics/zcoin

[69] ZILLIQA. (2017). *The ZILLIQA Technical Whitepaper [Version 0.1]*. Accessed: Nov. 8, 2022. [Online]. Available: https://docs.zilliqa.com/whitepaper.pdf

[70] H. T. M. Gamage, H. D. Weerasinghe, and N. G. J. Dias, "A survey on blockchain technology concepts, applications, and issues," *Social Netw. Comput. Sci.*, vol. 1, no. 2, p. 114, Mar. 2020, doi: 10.1007/s42979-020-00123-0.

[71] J. Fu, S. Qiao, Y. Huang, X. Si, B. Li, and C. Yuan, "A study on the optimization of blockchain hashing algorithm based on PRCA," *Secur. Commun. Netw.*, vol. 2020, pp. 1–12, Sep. 2020, doi: 10.1155/2020/8876317.

[72] J. R. Vacca, *Cyber Security and IT Infrastructure Protection*. Amsterdam, The Netherlands: Elsevier, 2014.

[73] I. Giechaskiel, C. Cremers, and K. B. Rasmussen, "When the crypto in cryptocurrencies breaks: Bitcoin security under broken primitives," *IEEE Secur. Privacy*, vol. 16, no. 4, pp. 46–56, Jul. 2018, doi: 10.1109/MSP.2018.3111253.

[74] A Tilooby, "The impact of blockchain technology on financial transactions," Ph.D. thesis, J. Mack Robinson College Bus., Georgia State Univ., Atlanta, GA, USA, 2018.

[75] *Secure Hash Standards (SHS)*, Standards FIPS180-4, Gaithersburg, MD, USA, 2015, doi: 10.6028/NIST.FIPS.180-4.

[76] I. Giechaskiel, "An evaluation of the effects of broken cryptographic primitives on bitcoin," Univ. Oxford, Oxford, U.K., Tech. Paper 27/15, 2015.

[77] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, "Demonstration of a small programmable quantum computer with atomic qubits," *Nature*, vol. 536, no. 7614, pp. 63–66, Aug. 2016, doi: 10.1038/nature18648.

[78] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[79] Etherscan. *Etherscan Network Hash Rate Chart*. Accessed: Mar. 1, 2023. [Online]. Available: https://etherscan.io/chart/hashrate

[80] INNOSILICON Technology Ltd. *Innosilicon*. Accessed: Apr. 3, 2023. [Online]. Available: https://innosilicon.shop/html/a12-miner/a12-plus-ltc-master.html

[81] COINTOMINE. *NiceHash SCrypt Mining Information*. Accessed: Mar. 23, 2023. [Online]. Available: https://cointomine.today/calculator/nicehash/scrypt/

[82] MinerStat. *NiceHash X11Pool Hash Rate*. Accessed: Mar. 16, 2023. [Online]. Available: https://minerstat.com/coin/NH-X11/network-hashrate

[83] MinerStat. *Equihash Hash Rate Chart*. Accessed: Mar. 15, 2023. [Online]. Available: https://minerstat.com/algorithm/equihash/charts

[84] S. Gupta, "Implementation of blockchain technology in supply chain," M.S. thesis, Dept. Logistics, Molde Univ. College, Molde, Norway, 2018.

[85] V. G. Martínez, L. Hernández-Álvarez, and L. H. Encinas, "Analysis of the cryptographic tools for blockchain and bitcoin," *Mathematics*, vol. 8, no. 1, pp. 1–14, Jan. 2020, doi: 10.3390/math8010131.

[86] V. S. Miller, "Use of elliptic curves in cryptography," in *Proc. Conf. Theory Appl. Cryptograph. Techn.*, Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 218. Berlin, Germany: Springer, 1986, pp. 417–426, doi: 10.1007/3-540-39799-X_31.

[87] D. R. L. Brown, "Standards for efficient cryptography 2 (SEC 2: Recommended elliptic curve domain parameters, Version 2.0)," Standard Efficient Cryptogr. Group (SECG), Mississauga, ON, Canada, 2010.

[88] B. K. Kikwai, "Elliptic curve digital signatures and their application in the bitcoin crypto-currency transactions," *Int. J. Sci. Res. Publication*, vol. 7, no. 11, pp. 135–138, 2017.

[89] P. M. S. Shoba, "A survey on post quantum digital signature schemes for blockchain," *Int. J. Comput. Sci. Mob. Comput.*, vol. 8, no. 6, pp. 128–133, 2019.

[90] S. D. Galbraith. *CRYPTREC Review of EdDSA*. Accessed: Jan. 17, 2023. [Online]. Available: https://www.cryptrec.go.jp/exreport/cryptrec-ex-3003-2020.pdf

[91] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, "High-speed high-security signatures," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, in Lecture Notes in Computer Science, vol. 6917, 2011, pp. 124–142, doi: 10.1007/978-3-642-23951-9_9.

[92] S. Josefsson and I. Liusvaara, *Edwards-Curve Digital Signature Algorithm (EdDSA)*, document RFC 8032, Internet Research Task Force (IRTF), 2017. [Online]. Available: https://www.rfc-editor.org/rfc/pdfrfc/rfc8032.txt.pdf

[93] G. M. Zaverucha and D. R. Stinson, "Short one-time signatures," Cryptol. ePrint Arch., 2010. Accessed: Jan. 17, 2023. [Online]. Available: https://eprint.iacr.org/2010/446.pdf

[94] Y. Xiao, P. Zhang, and Y. Liu, "Secure and efficient multi-signature schemes for fabric: An enterprise blockchain platform," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1782–1794, 2021, doi: 10.1109/TIFS.2020.3042070.

[95] M. Kara, A. Laouid, and M. Hammoudeh, "An efficient multi-signature scheme for blockchain," Cryptol. ePrint Arch., 2023. Accessed: Apr. 20, 2023. [Online]. Available: https://eprint.iacr.org/2023/078

[96] F. Shahid, I. ahmad, M. Imran, and M. Shoaib, "Novel one time signatures (NOTS): A compact post-quantum digital signature scheme," *IEEE Access*, vol. 8, pp. 15895–15906, 2020, doi: 10.1109/aCCESS.2020.2966259.

[97] T. M. Fernández-Caramès and P. Fraga-Lamas, "Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks," *IEEE Access*, vol. 8, pp. 21091–21116, 2020, doi: 10.1109/ACCESS.2020.2968985.

[98] W. A. A. Torres et al., "Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (Lattice RingCT v1.0)," in *Information Security and Privacy*. New York, NY, USA: Springer, 2018, pp. 558–576, doi: 10.1007/978-3-319-93638-3_32.

[99] T. G. Tan and J. Zhou, "A survey of digital signing in the post quantum era," Cryptol. ePrint Arch., 2019. Accessed: Jun. 22, 2023. [Online]. Available: https://eprint.iacr.org/2023/078.pdf

[100] Y.-L. Gao, X.-B. Chen, Y.-L. Chen, Y. Sun, X.-X. Niu, and Y.-X. Yang, "A secure cryptocurrency scheme based on post-quantum blockchain," *IEEE Access*, vol. 6, pp. 27205–27213, 2018, doi: 10.1109/ACCESS.2018.2827203.

[101] T. Liu, Y. Yuan, and Z. Yu, "The service architecture of Internet of Things terminal connection based on blockchain technology," *J. Supercomput.*, vol. 77, no. 11, pp. 12690–12710, Nov. 2021, doi: 10.1007/s11227-021-03774-9.

[102] P. Mccorry, "Applications of the blockchain using cryptography," Ph.D. thesis, School Comput., Newcastle Univ., Tyne, U.K., 2018.

[103] D. Di Francesco Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *Proc. IFIP Int. Conf. Distrib. Appl. Interoperable Syst.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 10320, 2017, pp. 206–220, doi: 10.1007/978-3-319-59665-5_15.

[104] X. Fu, H. Wang, and P. Shi, "A survey of blockchain consensus algorithms: Mechanism, design and applications," *Sci. China Inf. Sci.*, vol. 64, no. 2, Feb. 2021, Art. no. 121101, doi: 10.1007/s11432-019-2790-1.

[105] X. Zheng, Y. Zhu, and X. Si, "A survey on challenges and progresses in blockchain technologies: A performance and security perspective," *Appl. Sci.*, vol. 9, no. 22, pp. 1–24, Nov. 2019, doi: 10.3390/app9224731.

[106] M. Salimitari and M. Chatterjee, "A survey on consensus protocols in blockchain for IoT networks," Sep. 2018, pp. 1–15, *arXiv:1809.05613*.

[107] P. Hooda. (2022). *Blockchain—Proof of Work (PoW)*. Accessed: Jan. 22, 2022. [Online]. Available: https://www.geeksforgeeks.org/blockchain-proof-of-work-pow/

[108] H. Anwar. (2018). *Consensus Algorithms: The Root of Blockchain Technology*. Accessed: Dec. 25, 2021. [Online]. Available: https://101blockchains.com/consensus-algorithms-blockchain/

[109] 101 Blockchains. (2019). *Know Everything About Blockchain Proof Of Work (PoW)*. Accessed: Jun. 14, 2021. [Online]. Available: https://101blockchains.com/blockchain-proof-of-work/

[110] Q. Wang, J. Huang, S. Wang, Y. Chen, P. Zhang, and L. He, "A comparative study of blockchain consensus algorithms," *J. Phys., Conf. Ser.*, vol. 1437, no. 1, Jan. 2020, Art. no. 012007, doi: 10.1088/1742-6596/1437/1/012007.

[111] S. King and S. Nadal, "PPCoin: Peer-to-peer crypto-currency with proof-of-stake," 2012. Accessed: Jun. 18, 2022. [Online]. Available: https://decred.org/research/king2012.pdf

[112] NXT. (2016). *Nxt Whitepaper*. Accessed: Jan. 13, 2023. [Online]. Available: https://nxtdocs.jelurida.com/Nxt_Whitepaper

[113] V. Chawla. (2020). *What Are the Top Blockchain Consensus Algorithms?* Accessed: Mar. 29, 2023. [Online]. Available: https://analyticsindiamag.com/blockchain-consensus-algorithms/

[114] F. Yang, W. Zhou, Q. Wu, R. Long, N. N. Xiong, and M. Zhou, "Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism," *IEEE Access*, vol. 7, pp. 118541–118555, 2019, doi: 10.1109/ACCESS.2019.2935149.

[115] Dantheman. (2016). *DPOS Consensus Algorithm*. Accessed: Mar. 22, 2023. [Online]. Available: https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper

[116] Y. Sun, B. Yan, Y. Yao, and J. Yu, "DT-DPoS: A delegated proof of stake consensus algorithm with dynamic trust," *Proc. Comput. Sci.*, vol. 187, pp. 371–376, Jan. 2021, doi: 10.1016/j.procs.2021.04.113.

[117] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake [Extended Abstract]y," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 3, pp. 34–37, Dec. 2014, doi: 10.1145/2695533.2695545.

[118] M. Castro and B. L. Laboratory, "Practical Byzantine fault tolerance," in *Proc. 3rd Symp. Oper. Syst. Design Implement.*, New Orleans, LA, USA, Feb. 1999, pp. 1–14.

[119] Y. Li, L. Qiao, and Z. Lv, "An optimized byzantine fault tolerance algorithm for consortium blockchain," *Peer-Peer Netw. Appl.*, vol. 14, no. 5, pp. 2826–2839, Sep. 2021, doi: 10.1007/s12083-021-01103-8.

[120] S. Demirkan, I. Demirkan, and A. McKee, "Blockchain technology in the future of business cyber security and accounting," *J. Manage. Anal.*, vol. 7, no. 2, pp. 189–208, Apr. 2020, doi: 10.1080/23270012.2020.1731721.

[121] J. Sedlmeir, H. U. Buhl, G. Fridgen, and R. Keller, "The energy consumption of blockchain technology: Beyond myth," *Bus. Inf. Syst. Eng.*, vol. 62, no. 6, pp. 599–608, Dec. 2020, doi: 10.1007/s12599-020-00656-x.

[122] D. Schwartz, N. Youngs, and A. Britto. (2018). *The Ripple Protocol Consensus Algorithm*. Accessed: Apr. 1, 2023. [Online]. Available: https://ripple.com/files/ripple_consensus_whitepaper.pdf

[123] B. Chase and E. MacBrough, "Analysis of the XRP ledger consensus protocol," Feb. 2018, pp. 1–25, *arXiv:1802.07242*, doi: 10.48550/arXiv.1802.07242.

[124] *Direct Acyclic Graph Tangle (DAG)*. Accessed: Jan. 31, 2023. [Online]. Available: https://tokens-economy.gitbook.io/consensus/chain-based-dag/direct-acyclic-graph-tangle-dag

[125] B. Baek and J. Lin. (2019). *IOTA: A Cryptographic Perspective*. Accessed: Jan. 13, 2023. [Online]. Available: http://linpossible.com/papers/IOTA.pdf

[126] *Consensus in the IOTA Tangle*. Accessed: Dec. 25, 2022. [Online]. Available: https://blog.iota.org/consensus-in-the-iota-tangle-fpc-b98e0f1e8fa/

[127] M. Conti, G. Kumar, P. Nerurkar, R. Saha, and L. Vigneri, "A survey on security challenges and solutions in the IOTA," *J. Netw. Comput. Appl.*, vol. 203, Jul. 2022, Art. no. 103383, doi: 10.1016/j.jnca.2022.103383.

[128] Y. Li, B. Cao, M. Peng, L. Zhang, L. Zhang, D. Feng, and J. Yu, "Direct acyclic graph-based ledger for Internet of Things: Performance and security analysis," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1643–1656, Aug. 2020, doi: 10.1109/TNET.2020.2991994.

[129] W. Liang and N. Ji, "Privacy challenges of IoT-based blockchain: A systematic review," *Cluster Comput.*, vol. 25, no. 3, pp. 2203–2221, Jun. 2022, doi: 10.1007/s10586-021-03260-0.

[130] N. Raddatz, J. Coyne, P. Menard, and R. E. Crossler, "Becoming a blockchain user: Understanding consumers' benefits realisation to use blockchain-based applications," *Eur. J. Inf. Syst.*, vol. 32, no. 2, pp. 287–314, Mar. 2023, doi: 10.1080/0960085x.2021.1944823.

[131] S. Dhar and I. Bose, "Securing IoT devices using zero trust and blockchain," *J. Org. Comput. Electron. Commerce*, vol. 31, no. 1, pp. 18–34, Jan. 2021, doi: 10.1080/10919392.2020.1831870.

[132] R. Henry, A. Herzberg, and A. Kate, "Blockchain access privacy: Challenges and directions," *IEEE Secur. Privacy*, vol. 16, no. 4, pp. 38–45, Jul. 2018, doi: 10.1109/MSP.2018.3111245.

[133] H. Nejatollahi, N. Dutt, S. Ray, F. Regazzoni, I. Banerjee, and R. Cammarota, "Post-quantum lattice-based cryptography implementations," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–41, Nov. 2019, doi: 10.1145/3292548.

[134] D.-T. Dam, T.-H. Tran, V.-P. Hoang, C.-K. Pham, and T.-T. Hoang, "A survey of post-quantum cryptography: Start of a new race," *Cryptography*, vol. 7, no. 3, p. 40, Aug. 2023, doi: 10.3390/cryptography7030040.

[135] S. Balogh, O. Gallo, R. Ploszek, P. Špač ek, and P. Zajac, "IoT security challenges: Cloud and blockchain, postquantum cryptography, and evolutionary techniques," *Electronics*, vol. 10, no. 21, p. 2647, Oct. 2021, doi: 10.3390/electronics10212647.

[136] A. Sayakkara, N.-A. Le-Khac, and M. Scanlon, "A survey of electromagnetic side-channel attacks and discussion on their case-progressing potential for digital forensics," *Digit. Invest.*, vol. 29, pp. 43–54, Jun. 2019, doi: 10.1016/j.diin.2019.03.002.

[137] M. Devi and A. Majumder, "Side-channel attack in Internet of Things: A survey," in *Applications of Internet of Things* (Lecture Notes in Networks and Systems), vol. 137, 2021, pp. 213–222, doi: 10.1007/978-981-15-6198-6_20.

[138] E. Prouff and M. Rivain, "Masking against side-channel attacks: A formal security proof," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, in Lecture Notes in Computer Science, vol. 7881, 2013, pp. 142–159, doi: 10.1007/978-3-642-38348-9_9.

[139] A. Kerckhoffs, "La cryptographie militaire—Seconde part," *J. des Sci. Militaires*, 1883. Accessed: May 12, 2023. [Online]. Available: https://www.petitcolas.net/kerckhoffs/crypto_militaire_2.pdf

[140] O. Pal, B. Alam, V. Thakur, and S. Singh, "Key management for blockchain technology," *ICT Exp.*, vol. 7, no. 1, pp. 76–80, Mar. 2021, doi: 10.1016/j.icte.2019.08.002.

[141] J. B. Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. T. Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: Review and challenges," *IEEE Access*, vol. 7, pp. 164908–164940, 2019, doi: 10.1109/ACCESS.2019.2950872.

[142] A. Huang, S. Barz, E. Andersson, and V. Makarov, "Implementation vulnerabilities in general quantum cryptography," *New J. Phys.*, vol. 20, no. 10, Oct. 2018, Art. no. 103016, doi: 10.1088/1367-2630/aade06.

[143] S. Schulte, M. Sigwart, P. Frauenthaler, and M. Borkowski, "Towards blockchain interoperability," in *Proc. Int. Conf. Bus. Process Manag.*, in Lecture Notes in Business Information Processing, vol. 361, 2019, pp. 3–10, doi: 10.1007/978-3-030-30429-4_1.

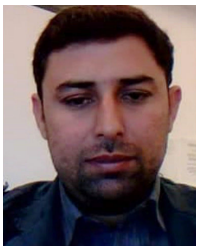
[144] *Contingency Plans—Bitcoin Wiki*. Accessed: Oct. 18, 2023. [Online]. Available: https://en.bitcoin.it/wiki/Contingency_plans

[145] I. Giechaskiel, C. Cremers, and K. B. Rasmussen, "When the crypto in cryptocurrencies breaks: Bitcoin security under broken primitives," *IEEE Secur. Privacy*, vol. 16, no. 4, pp. 46–56, Jul. 2018, doi: 10.1109/MSP.2018.3111253.

[146] M. I. Sarwar, L. A. Maghrabi, K. Nisar, and I. Khan, "Cryptovirology ransomware: A review of dissemination and mitigation techniques," *Inf. Sci. Lett.*, vol. 12, no. 11, pp. 2277–2288, 2023, doi: 10.18576/isl/121135.

**MUHAMMAD IMRAN SARWAR** (Member, IEEE) received the bachelor's (B.Com.) degree in advanced accounting and auditing from the University of the Punjab, Lahore, Pakistan, in 1996, and the master's degree in information technology (IT) from the Department of Computer Science and IT, The Superior College, Lahore, in 2021. He is currently pursuing the Ph.D. degree in computer science with the Department of Computer Science and IT, Superior University, Lahore. Alongside his academic pursuits, he holds the position of the IT Head in a public sector organization, where he is responsible for overseeing all IT operations. His research interests include blockchain, database management systems (DBMS), enterprise resource planning (ERP), accounting information systems (AIS), management information systems (MIS), financial management systems (FMS), IT management (ITM), IT service management (ITSM), supply chain management (SCM), the Internet of Things (IoT), sensors and industry-specific specialized solutions.

**LOUAI A. MAGHRABI** (Member, IEEE) received the B.Sc. degree in computer science from Lebanese American University, Beirut, Lebanon, the M.Sc. degree in information technology from the University of West of England, Bristol, U.K., and the Ph.D. degree in cybersecurity from Kingston University, London, U.K. He is currently an Assistant Professor with the Department of Software Engineering, College of Engineering, University of Business and Technology, Jeddah, Saudi Arabia. His research interests include cybersecurity, risk assessment, cryptography, artificial intelligence, machine learning, the IoT, blockchain, drones, metaverse, quantum computing, and game theory. He received the best research paper award in 2021.



**IMRAN KHAN** received the Ph.D. degree in computer science from the National University of Computer and Emerging Sciences, Islamabad, Pakistan, in 2018. He was a Software Developer and a Researcher with the University of Birmingham, U.K., and the Security Engineering Research Group (SERG) IMSciences, Pakistan, on cloud and web security-related domains. He is currently an Assistant Professor with the Department of Computer Science, Superior University, Lahore, Pakistan. He is also a Postdoctoral Research Fellow with the Department of Computer and Informatics, Cardiff University, U.K. His current research interests include cloud computing, distributed systems security, and blockchain technology.

**QAMAR H. NAITH** (Member, IEEE) received the B.Sc. degree in computer science from Umm Al-Qura University, Makkah, Saudi Arabia, in 2010, the M.Sc. degree in computer science from Toronto Metropolitan University, Toronto, Canada, in 2015, and the Ph.D. degree in software engineering from The University of Sheffield, Sheffield, U.K., in 2021. She is currently an Assistant Professor with the Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia. She has gained more than seven certificates in the different subjects of patents in the technology field. Her current research interests include the Internet of Things (IoT), mobile computing, human–computer interaction (HCI), user interface and user experience (UI/UX), machine learning (ML), drones, blockchain, artificial intelligence (AI), and AI technologies to improve HCI and the medical field. In 2020, she was awarded for contributing to developing a system that helps heart disease patients. She was also awarded the Best Student Research Paper Award, in 2018, and the Best Ph.D. Researcher Award in software testing, in 2021.

**KASHIF NISAR** (Member, IEEE) received the master's degree in information technology (IT) from the Department of Computer Science and IT, The Superior College, Lahore, Pakistan, in 2021. He is currently pursuing the Ph.D. degree in computer science with the Department of Computer Science and IT, Superior University, Lahore. He is a Senior Network Engineer with Kind Edward Medical University, Lahore. His research interests include IoT, blockchain, machine learning (ML), artificial intelligence (AI), network and hardware, human-computer interaction, healthcare management systems, cryptography, and industry-based specialized solutions.