ISSN 2723-6471

Security Issues and Weaknesses in Blockchain Cloud Infrastructure: A Review Article

Hala A. Albaroodi[1,*], Mohammed Anbar[2]

[1]*Gifted guardianship committee, in Ministry of Education, 10001 Aljamiea, Baghdad, Iraq*

[2]*National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia (USM), 11800 Gelugor, Penang, Malaysia*

(Received: July 17, 2024; Revised: September 5, 2024; Accepted: November 15, 2024; Available online: December 28, 2024)

Abstract

Cloud computing has become an essential technology due to its ability to provide scalable infrastructure and data services at a low cost and with minimal effort. It is widely adopted across various IT sectors and excels in providing flexible and scalable solutions for storage, computation, and networking. However, despite its widespread adoption, information security concerns remain a significant challenge, hampering its full potential. Issues such as data breaches, insufficient access controls, privacy risks, and vulnerability to external attacks persist, making security a critical obstacle for cloud computing's growth. At the same time, blockchain technology has emerged as a promising solution for addressing these security challenges. Celebrated for ensuring data integrity, authenticity, and confidentiality, blockchain's decentralized structure offers a potential safeguard against the risks cloud systems face. For instance, blockchain's ability to maintain an immutable, tamper-proof ledger and decentralized control can mitigate unauthorised access risks, thereby enhancing cloud environments' transparency and security. One of the blockchain's core components is the consensus protocol, a method through which a network of nodes validates transactions without needing to trust any single entity. In the case of Bitcoin, users follow the Proof of Work algorithm, dedicating hardware and energy resources to solve cryptographic puzzles and verify transactions. This decentralized verification process addresses fraud concerns, but it also brings challenges such as high energy consumption and network centralization, particularly in regions with cheap electricity. These concerns have led to worries about collusion risks and policy changes affecting the stability of the network. Blockchain's decentralized nature has sparked significant interest, especially in its potential to enhance cloud computing security. Its ability to provide tamper-proof transaction logs, eliminate single points of failure, and grant users more control over data aligns well with the security demands of cloud environments. However, blockchain itself faces challenges, including scalability issues and its association with black-market trading due to its open-access model. Despite these concerns, blockchain's integration into cloud systems presents a unique opportunity for addressing key security obstacles, thereby offering more robust solutions for corporate and financial applications.

Keywords: Security, Blockchain, Cloud, Bitcoin, Decentralization, Encrypted, Decrypted, Transactions, Cryptography, Consensus Protocol, Hash Function

1. Introduction

Open Source Cloud Computing (OSCC) has brought numerous benefits in both industrial and academic environments. It has facilitated cost-effective solutions, enhanced flexibility, and encouraged innovation. However, it is essential to acknowledge that OSCC also introduces potential risks such as data security concerns, compliance issues, and reliance on community support. Therefore, while embracing OSCC, it is essential to carefully assess and address these risks to ensure the overall stability and security of the system [1]. The data belonging to users of cloud services faces a substantial risk of being lost, hacked, or compromised, leaving them with no means to safeguard themselves from this precarious situation. Cloud users are often unaware of the entities they are engaging with or sharing their information with, further exacerbating their vulnerability to potential threats.

Transparency is an essential issue for cloud customers. The lack of visibility into data access and movement within the cloud poses significant risks that cloud providers must address and mitigate [1], [2] with sufficient understanding between cloud consumers or organizations. Blockchain is a growing technology cloud customers can utilize to increase trust and data security while outsourcing and acquiring cloud services. When compared to the security of a centralized database, Blockchain can provide significant protection [3]. In the world of blockchain technology, effective communication is crucial for the creation and validation of new transactions. Furthermore, reaching a consensus on the

***Corresponding author: Hala A. Albaroodi (hala.albaroodi5@gmail.com)**

DOI: https://doi.org/10.47738/jads.v6i1.324

ledger, which serves as a verified transaction record, also requires seamless communication [4]. This communication occurs among nodes, each maintaining a copy of the ledger and informing others about newly submitted or confirmed transactions. Administrators of private blockchains have the authority to control who can run a node and how these nodes are interconnected. Notably, nodes with a higher number of connections receive information at a faster rate [5].

Moreover, nodes are often expected to maintain a specific number of connections to be considered active. Identifying and avoiding nodes that impede information transfer or transmit inaccurate data is imperative to ensure the system's integrity. In the context of a private blockchain underpinning commodities trade, trusted trading partners may hold more central positions within the network. As a security measure, newly added nodes may be required to maintain a connection to one of these significant nodes [6]. The significance of presenting apparent novelty in the article can be overstated, as it is paramount to avoid the risk of sounding like a mere summary of existing knowledge while contributing new insights.

2. Cloud Computing

Cloud Computing is a methodology for providing on-demand network access to a shared pool of programmable computing resources that can be quickly supplied and released with minimum administration effort and service provider contact [1]. Different services are given to the end-user using three different delivery methods in Cloud Computing. Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) are three delivery models that give infrastructure resources, application platforms, and software as services to consumers [2]. Figure 1 depicts the strong association between Cloud Computing features, cloud model deployment, and service model deployment. The cloud environment is likewise subjected to varying levels of security requirements. All cloud services are built on the base of IaaS; PaaS builds on IaaS, and SaaS builds on PaaS. Information security challenges and dangers are passed down through consecutive models as they inherit capabilities [3]. Figure 1 shows the Cloud Computing architecture

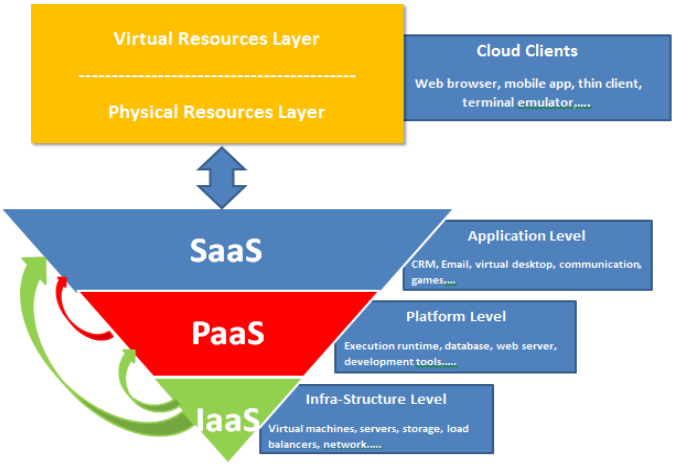
Virtual Resources Layer
Physical Resources Layer
Cloud Clients
Web browser, mobile app, thin client, terminal emulator,....
SaaS
Application Level
CRM, Email, virtual desktop, communication, games,...
PaaS
Platform Level
Execution runtime, database, web server, development tools.....
IaaS
Infra-Structure Level
Virtual machines, servers, storage, load balancers, network.....

Figure 1. Cloud Computing Architecture [3].

Cloud computing offers several advantages, making it highly attractive for individual users and organizations. Its scalability allows the network to support millions of users and nodes with a flexible hardware architecture that can scale in and out. Elasticity enables the system to adjust workloads by dynamically allocating and deallocating resources, ensuring that all resources are optimally used according to demand. Privacy is crucial, as it allows users to control their data securely. With infinite computing resources, users don't need to plan for capacity in advance, as cloud services provide resources on demand. Costs vary depending on applications and usage frequency, meaning users pay based on the extent of resource consumption. Utilization is maximized as resources can be adjusted to handle varying loads. Cost-effectiveness is another key advantage, with cloud services offered on-demand, eliminating the need for software licenses and hardware-specific software. Performance is critical, as high efficiency is essential for applications running on cloud systems. Finally, data sharing and access flexibility enhance user freedom, making cloud computing highly adaptable and user-friendly [4].

3. Ethereum

The header, transactions, including the transaction hash or transaction root, and the state radio booth, the state hash, or the state root, are the major components of an Ethereum block [5]. The block's integrity is maintained by ensuring that the block header contents are not tampered with, that transactions are not tampered with, and that state transitions are calculated, hashed, and verified effectively [6]. Keep in mind that the Blockchain is designed to be an unchangeable record. The block hash in Ethereum is the sum of all the items in the block header, including the hashes of the transaction root and state root [7]. As shown in figure 2, it computes using a variation of the SHA-3 method called Keccak and all of the block header components.

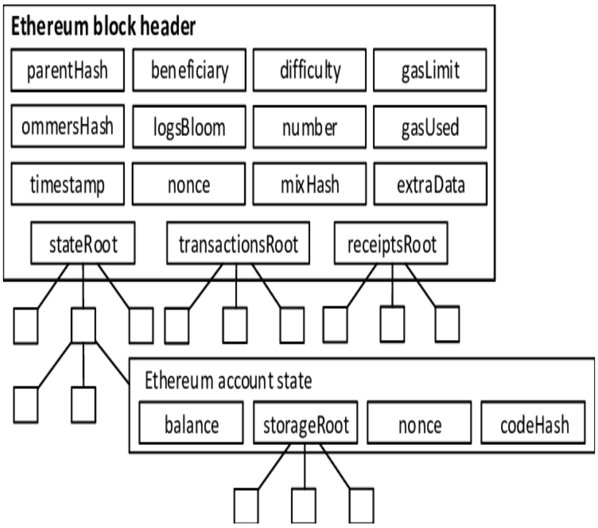

Figure 2. Ethereum Block Header [7].

On the other hand, bitcoin, a typical block, has roughly 2,000 transactions, whereas in Ethereum, a typical block contains about 100 transactions. The require a reliable method for detecting tampering and validating transactions [8]. The Merkle tree hash, which we examined in an earlier lesson, is used to process the hashes of transactions in a block. Because only the hash of the chained states from one block to the next must be recomputed, the Merkle tree hash is also employed to compute the state root hash. It's also utilized to determine the hash root of receipts. Keep in mind the benefits of flat vs tree representation [9]. In a Merkle tree, leaf nodes form the lowest layer, such as L1, L2, L3, and L4. Child nodes are nodes at lower tiers that contribute to a parent node's value; for example, "Hash 0-0" and "Hash 0-1" are the children of the "Hash 0" node. At the very top is the root node, the single node at the uppermost level, also known as the "Top Hash." Additionally, orphaned blocks may arise if excessive mining leads to the simultaneous creation of multiple blocks, leaving some blocks excluded from the main chain.

Thousands upon thousands of transactions are contained in each block. Keeping all the data in each block as a series will be inefficient. Finding any specific transaction will be extremely difficult and time-consuming if you do so. However, utilizing a Merkle tree may significantly reduce the time it takes to determine if a specific transaction fits in that block [10].

Rather than going through the time-consuming process of examining each hash to determine if it corresponds to the data, trace the trail of hashes leading up to it. This cut down on the amount of time it takes. When we mention "mining," we're looking for a new block to be put into the Blockchain [11]. Miners from all around the world are working hard to ensure that the chain continues to expand. People used to be able to mine with only their laptops, but as time went on, they began to join mining pools to pool their computer power and mine more effectively; nevertheless, this might have been an issue. Each cryptocurrency, for example, has a cap. There are only 21 million bitcoins. There are a total of 21 million bitcoins in circulation. If the miners keep going at this rate, they can extract all the existing bitcoins [12].

Mining is similar to a game in that you solve a puzzle and receive rewards. Setting the difficulty raises the complexity of the challenge, making it more time-consuming to solve. As a result, a predefined difficulty level is chosen to limit block construction. The difficulty goal for bitcoins is a 64-character string that starts with several zeroes (the same as a SHA-256 result). As the difficulty level rises, so does the quantity of zeros. After every 2016 block, the difficulty level changes [13].

Only one path to the tree must be checked if any transaction is to be confirmed. There's no need to look through every single transaction. State transitions occur as a result of Ethereum's intelligent contract execution. Every state change necessitates a re-computation of the state root hash. Only the damaged path in the Merkle tree has to be recomputed rather than the complete collection of states. When state 19 changes to 20, the route direction is altered to 31, 41, and the state root hash 64 recalculates. As seen at the bottom of the block header, the block hash is produced by computing the state root hash, transaction root hash, and receipt root hash [14]. Figure 3 shows that the path is recomputed, not the entire tree [10].

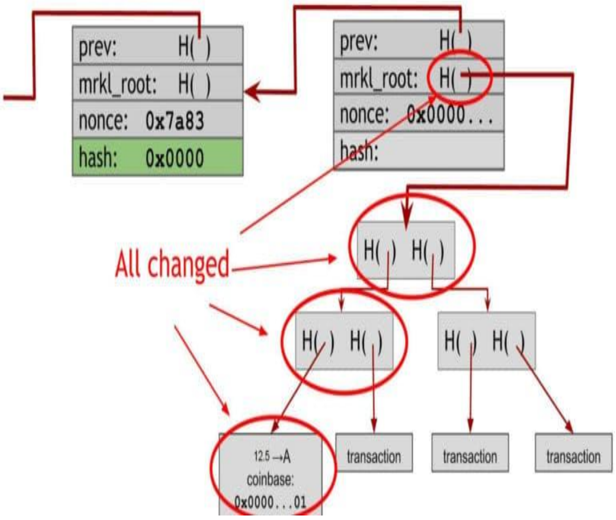

Figure 3. Merkle Tree [14].

To solve the PoW puzzle, these roots and the other elements in the header are hashed along with the variable nodes. By embedding the previous block hash in the current block header, the block hash provides two essential functions: verification of the block's integrity and transactions and establishment of the chain link [20]. Assume that any participating node tampers with the block, changing its hash value, causing a discrepancy in the hash values, and declaring the node's local chain invalid [15]. Due to a hash discrepancy, other miners would reject any future blocks initiated by the node, ensuring the chain's immutability. To summarize, the various pieces of the Blockchain are secured using a mix of hashing and encryption [16]. In decentralized networks that operate outside trust borders, the private-public key pair and hashing are crucial underlying principles [17], [18].

4. Blockchain

Through communication, blockchains achieve consensus on their ledger, which is the list of verified transactions. Communication is necessary to write and approve new transactions. This communication occurs between nodes, each of which maintains a copy of the ledger and informs the other nodes of new information, such as newly submitted or verified transactions. Operators of private blockchains can control who is allowed to operate a node, as well as how those nodes are connected. A node with more connections will receive information faster.

Similarly, nodes may be required to maintain a certain number of connections to be considered active. To maintain the integrity of the system, a node that restricts the transmission of information or transmits incorrect information must be identifiable and able to be circumvented. In a private blockchain for commodities trading, more established trading partners may be granted more central positions in the network. New nodes may be required to connect to one of these central nodes as a security measure to ensure proper behavior [19].

5. Blockchain Structures

A data structure is a specific type of data storage. Understanding how a Blockchain operates necessitates the knowledge of two data structure features [20], [21], [22]. They are as follows:

5.1. Pointers

In programming, pointers are variables that hold the address of another variable. In most programming languages, standard variables are used to store data. For example, Int a = 10 denotes the existence of a variable "a" that stores integer values. It holds an integer value of 10 in this example, a standard variable. Instead of storing values, pointers hold addresses for other variables, which is why they are called pointers; they point to other variables' locations [23], [24].

5.2. Linked Lists

A linked list is one of the most critical data structure components, and it looks like this: It consists of a series of blocks, each of which contains data and is connected to the next block by a pointer. In this situation, the pointer variable has the address of the next node, and therefore the connection is established. As you can see, the last node contains a null pointer, which means it has no value [24].

One thing to remember is that each block's pointer carries the location of the following block. This is how pointing is accomplished. The initial block is known as the "genesis block," its pointer is outside the system. The Blockchain is a linked list of data with a hash pointer to the preceding block, which forms the chain. A hash pointer is similar to a pointer, except instead of merely holding the previous block's location, it also contains the hash of the data within the previous block. This is a minor change that makes Blockchains so incredibly dependable and forward-thinking [25].

Assume a hacker has access to block three and attempts to alter the data. Because of the nature of hash functions, even a minor change in data will substantially alter the hash. This means that any minor changes made in block three will change the hash stored in block 2, which will change the data and hash of block two. Consequently, adjustments will be made to block one and subsequent blocks. This will cause the chain to alter, which is impossible [9], [31] totally. The immutability of blockchains is achieved. A block header contains essential information, including the block version number, which indicates the version of the block, the current date and time of the block's creation, and the current difficulty level associated with mining the block. It also includes the hash of the preceding block, linking it to the previous block in the chain; a nonce, a unique value used to satisfy the difficulty requirement for block mining; and the Merkle Root Hash, representing the combined hash of all transactions within the block.

5.3. SHA-3 Hash Function

The third type of cryptography is the hash function, which encrypts data using mathematical modeling. The Hash Function is a one-way function with no key since ciphertext cannot cover plaintext. In cryptography, the hash function is helpful for ensuring data validity, notably in computer network security. As communication technology improved, cryptographic security issues arose to secure the sender's data and receivers [26]. It has been utilized for various additional purposes since the discovery of secure hash functions [27]. Before delivering any information to the server, the authentication information is encrypted using a symmetric cryptography technique, and extra hash functions are utilized, as illustrated in figure 4.

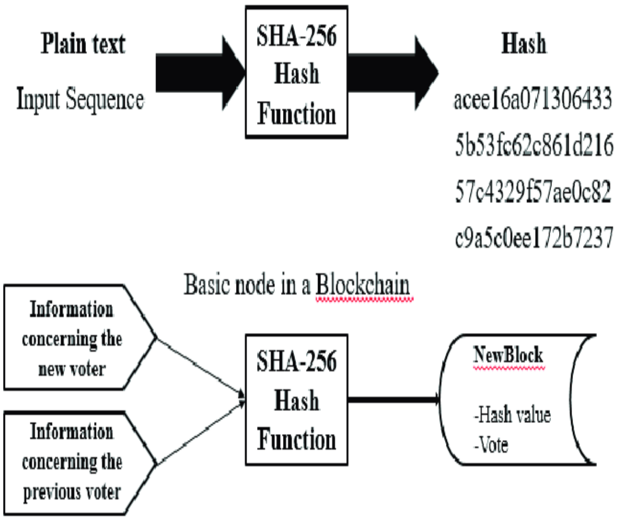

Figure 4. Hash Function SHA-256 [28].

The SHA hash function is a cryptographic function created by the National Security Agency (NSA) [29]. Secure Hash Algorithm (SHA) is an acronym for Secure Hash Algorithm. The three SHA algorithms are known as SHA-0, SHA-1, and SHA-2 and are organized differently. The SHA-1 algorithm can calculate a string's hash value. SHA-1, SHA-2, SHA-3, SHA-256, SHA-384, and SHA-512 are all viable Message Digest algorithms [30].

6. Blockchain Security Challenges

When creating network architecture, uncommunicative or occasionally active nodes are considered security risks [30]. Nodes can go offline for various reasons, but the network must be designed to function without them, and it must be able to swiftly bring them back up to speed if they do [31].

A single primary chain with a continuous state is called a secure chain. Every valid block contributed to this chain raises the chain's trust level. A competitive miner created each of the candidate blocks [31]. The method or process for selecting the next block is known as PoW. Hashing is used in PoW [32]. Calculate the hash of the block header components, a constant number, and a nonce, a variable. First, if the hash value is less than two pairs (128 for Bitcoin) and less than the difficulty function for Ethereum, the problem is solved. If it still hasn't been solved, repeat the process after adjusting the nonce value; if it still hasn't been solved, the riddle has been solved, and the winning block should be broadcast for other miners to verify [33], [34].

For value transfer platforms, cryptocurrencies employ distributed ledgers, or Blockchains, to store information about each address's balance. The method, however, may be used for any data. The Blockchain's operation is dependent on the network's collective agreement on the contents of the ledger: rather than having authority for keeping accounts centralized in one entity, such as a bank, it is shared amongst all; this necessitates the network's consensus on the information recorded on the Blockchain. In addition, reaching a consensus influences the protocol's security and economic factors [35], [36]. These consensus procedures are explained below:

6.1. Proof of Work (PoW)

The PoW distributed consensus process was invented by Satoshi Nakamoto, the mysterious person who created bitcoin. Many cryptocurrencies, including Ethereum, did the same. In PoW, all of the machines in the network responsible for ensuring the Blockchain's security, or "Miners" in the context of bitcoin, cooperate to solve a puzzle of a mathematical operation called a hash. Although easy for a machine to do, this operation requires a lot of repetition, which increases the computational cost. A hash with particular characteristics is sought after by computers [36], [37]. A new block of transactions may be added to the Blockchain by the computer that arrives at the solution first and provides supporting evidence that they have completed the required work. Figure 5 shows that they receive a portion of freshly created bitcoins (12.5 BTC every block, or about every 10 minutes), plus all of the modest transaction fees customers paid to transmit coins.

It is expensive to add a batch of new transactions to the Blockchain because of the transparent format of the ledger. Still, it is relatively simple to determine whether the transactions are legitimate. Transactions are not entirely "confirmed" until miners have added multiple blocks to the Blockchain, collectively validated by miners [38]. The remainder of the network will ignore any fraudulent coin spending attempts made by malicious actors. The only way an attacker could pull off such a scam is if they had a significant processing advantage, repeatedly beating the PoW competition using the "51 percent assault," which necessitates possessing more than half of the network's hash rate. No miner has ever been successful [39], As indicated in figure 5.

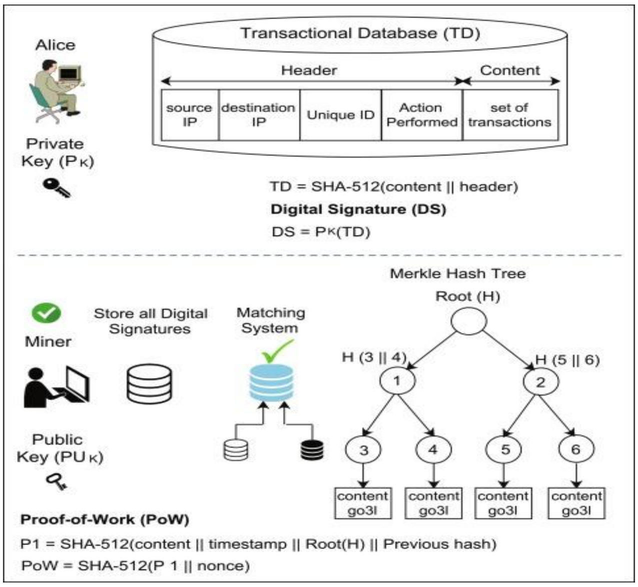

Figure 5. Proof of Work (PoW) [37].

6.2. Proof of Stake (PoS)

Specifically designed chips for mining have given rise to a whole industry. A more modern technique that doesn't need any special hardware has gained popularity recently: PoS. PoW costs money and uses energy since it requires so much processing. The probability that a participant will contribute the following block of transactions to the Blockchain depends on their hash rate in the PoW protocol. In PoS, a participant's coin stake affects their likelihood [36]. Each node in the network has an address assigned to it, and the more coins that address has, the more likely it is that the node will mine (or "stake") the following block. It is comparable to a lottery in that the winner is selected randomly, but the odds are better if they have more coins (lottery tickets) [39].

6.3. Leased Proof of Stake (LPoS)

Like tiny miners with low hash rates are unlikely to mine a block in bitcoin, conventional PoS holders with low balances are unlikely to stake a block. Several years may pass before a smallholder is fortunate enough to develop a block. As a result, fewer key participants are left to operate the network since many holders with low balances choose not to run a node. Motivating these smaller holders to engage is major since network security improves with more members [39].

Waves use the LPoS method. Leasing balances to staking nodes is one way LPoS does this [1], [40]. The leased monies are still entirely under the holder's control and may be relocated or spent whenever desired (at which point the lease expires) [41]. Leased coins raise the staking node's "weight," which raises the probability that it will be granted permission to add a block of transactions to the Blockchain. Better chances and a proportionate part of any gains are given to the lessees [42]. An attacker would require more than half of the coins to complete the necessary transactions effectively; buying these would raise the cost and make such a task prohibitively costly [40]. Each of the two variations, LPoS and DPoS, has unique advantages. It doesn't need cost recovery in the same way as bitcoin does since it is less energy-intensive than PoW. Therefore, platforms with a fixed currency supply and no block reward inflation are most suited for PoS systems. Stakers would not support the method used by most crowd-funded platforms, which issue tokens depending on investment and dilute them with excess currencies [43].

6.4. Delegated Proof of Stake (DPoS)

Currency owners can vote on a list of nodes, which allows them to stake new transaction blocks and add them to the Blockchain using their balances under DPoS. Even while it might not instantly reward them in the same way as LPoS does, this affects all currency holders. Holders have more power and influence over the network because of the ability to vote on changes to network settings. BitShares uses a similar but somewhat different method [40].

6.5. Proof of Importance (PoI)

The last phase of these consensus procedures is called Proof of PoI [44]. NEM, the "New Economy Movement," was the first cryptocurrency platform to take this action. When it comes to PoI, it's not only about coined balance [40]. The foundation of NEM's consensus algorithm is that useful network activity should be rewarded rather than the quantity of money. Balance, reputation (which is determined by a different specially developed algorithm), and the volume of transactions to and from that address are some factors that influence whether a block gets staked. PoI provides a fuller picture of a "valuable" network participant.

Numerous platforms combine PoW and PoS; the first is used to distribute funds before switching to the second to maintain the network. There are many variations of these fundamental ideas. Master-nodes combined with PoW mining is a further method, as seen with DASH and Crown. These aid in transaction processing and receive a share of the benefits from blocks miners create through their work. Regardless of the situation, the consensus approach works to protect the network, primarily through economic means: attempting to attack the network should be prohibitively expensive while assisting in its protection should be more valuable [45].

7. Public Security Satisfaction

Customers have high expectations for both the efficient execution of tasks and the swift and effective resolution of any mistakes. In this context, robustness refers to a system's ability to continue functioning under adverse conditions or when encountering unexpected disruptions. This is particularly crucial in decentralized networks like Blockchain, where no single entity manages the data or operations [46].

The subsequent section will detail the specific challenges that can occur in Blockchain processes. For Blockchain to maintain its security, it must be capable of handling variations in network performance, potential system failures, and security threats without compromising the integrity of the ledger. Subsequent sections will delve deeper into these specific challenges and outline how they can be effectively addressed. All of these issues will be covered in the following sections:

7.1. Trust

Recognize that on a decentralized Blockchain, trustworthiness also comprises securing, verifying, confirming, and ensuring the availability of resources needed for transaction execution [47], [48]. This is accomplished by safeguarding the chain with the proper protocols, verifying transactions and blocks for tamper-proofing, determining whether resources are available for commerce, and executing and confirming transactions, as shown in figure 6. The Trust Trail involves a series of actions to maintain the integrity of a blockchain system. First, transaction validation ensures each transaction meets necessary requirements. Then, resource and gas availability are verified to confirm sufficient resources for transaction processing. An inventory of transactions is created, followed by transaction completion to establish a new state. Next, a new block is added, and efforts are made to reach consensus among network participants. Finally, the chain expands as new blocks are added, and transactions within these blocks undergo verification to ensure accuracy and consistenc.

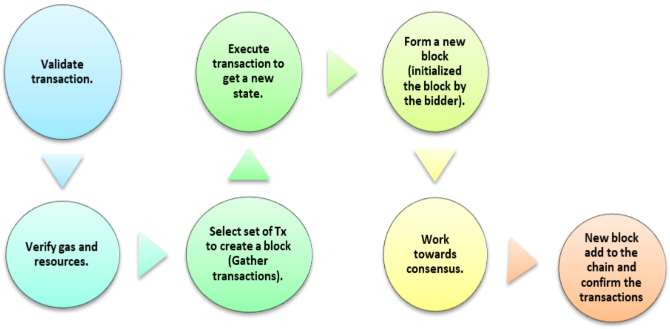
Validate transaction.
Execute transaction to get a new state.
Form a new block (initialized the block by the bidder).
Verify gas and resources.
Select set of Tx to create a block (Gather transactions).
Work towards consensus.
New block add to the chain and confirm the transactions

Figure 6. Trust Operations.

Examine each of these actions. The first and second phases confirm the resources and verify the transaction, respectively. In the case of Bitcoin, there are roughly 20 criteria that must be checked before a transaction is valid (the syntax, the transaction signature, timestamp, nonce, gas limit, and sender account balance are all verified before execution); similarly, in the case of Ethereum transactions, the availability of fuel, gas stations, and other resources for flawless contract implementation is also confirmed. Transaction hashes and signatures are also verified. Transactions are carried out in the third stage. The Merkle tree hash of the validated transactions is computed. All miners participate in the transaction for the transfer or execution of smart contracts. The state resulting from transaction execution is used to compute the Merkle tree hash of the states, which is the state root of the block header. Also determined is the reception root of the block header [39], [1].

Trust mechanisms in blockchain networks rely on decentralized models that eliminate the need for a central authority by enabling a distributed network of participants to validate and secure transactions. Different blockchain networks implement these mechanisms in various ways, using consensus algorithms to maintain trust and prevent malicious activity. It would be interesting to explore examples of how decentralized trust models operate across different blockchain networks and their security implications [49].

7.1.1. Proof of Work (PoW)

PoW Utilized by Bitcoin and Ethereum (pre-2022). In PoW, participants (miners) compete to solve complex mathematical puzzles to validate transactions and create new blocks. The first miner to solve the puzzle earns the right to append the block to the chain and is rewarded with cryptocurrency [50].

Security Implications in PoW are considered secure due to the high computational power required to manipulate the network (51% attack), where a malicious actor would need to control more than half of the network's hashing power. The drawbacks of PoW include high energy consumption and the risk of network centralization, as mining tends to concentrate in regions with cheaper electricity. This could lead to potential collusion and control by a few entities [51], [52].

7.1.2. Proof of Stake (PoS)

They were utilized by Ethereum 2.0, Cardano, and Polkadot. PoS selects validators to create new blocks and validate transactions based on the amount of cryptocurrency they "stake" as collateral. The more coins a participant stakes, the higher their chance of being chosen to validate transactions [53].

Security Implications in PoS are more energy-efficient than PoW as it does not require intensive computational work. It enhances security by penalizing malicious actors who stake their coins, ensuring they have "skin in the game." If validators behave dishonestly, they lose their staked assets. The Drawbacks with PoS Wealthier participants with more staked coins have a higher chance of controlling the network, potentially leading to centralization.

7.1.3. Delegated Proof of Stake (DPoS)

EOS, and TRON utilized DPoS. In DPoS, token holders select a small group of delegates or witnesses accountable for validating transactions and safeguarding the network. These delegates are motivated to prioritize the community's best interests to maintain their position [54].

Due to the fewer nodes involved in validation, security Implications in DPoS offer superior speed and efficiency compared to PoW and PoS. The Drawbacks with DPoS The voting process has the potential to result in centralization, as a small group of delegates may wield disproportionate control over the network, giving rise to concerns about collusion and diminished decentralization [54].

7.1.4. Byzantine Fault Tolerance (BFT)

This text provides valuable insights into the BFT consensus mechanism, which prominent platforms such as Hyperledger, Cosmos, and Ripple utilize. BFT models ensure that the network can still reach consensus even if some nodes act maliciously or fail, as long as most nodes remain honest. The Practical Byzantine Fault Tolerance (PBFT) variant enables participants to agree on the Blockchain's state through communication between nodes, often with a known validator set. Regarding security implications, BFT models offer robust resistance to malicious behavior, as attacks would require a significant proportion of nodes to be compromised. However, BFT may encounter scalability challenges as the network grows, leading to increased communication overhead. Furthermore, the reliance on a known validator set reduces certain aspects of decentralization [55].

7.1.5. Proof of Authority (PoA)

VeChain, some Ethereum sidechains utilize them. Mechanism: In PoA, trusted validators are pre-approved by the network to validate transactions based on their reputation. Validators are typically known and reputable entities. Security Implications: PoA networks offer high throughput and low latency, making them efficient for specific use cases such as private or permissioned blockchains. Nonetheless, PoA's trust model is more centralized because validators are known and approved, introducing reliance on trusted entities and diminishing the trustless and decentralized nature of the system [56], [57].

7.2. Integrity

In a decentralized network, participants must be able to identify one another uniformly. Let's start with the addresses of the accounts. A distinctive account address is necessary to preserve transaction integrity [39]. The transaction is authorized by the sender's digital signature, which comes in at number two, and the authenticity of the transaction's content comes in at number three [1]. Generating account addresses involves a sequence of steps using a public-private key pair. In the first step, a 256-bit random integer is created, forming the private key, which is securely stored and kept confidential. In the second step, an ECC algorithm is applied to the private key, generating a unique public key. Finally, in the third step, the public key undergoes hashing to produce the account address, a shorter identifier of 20 bytes (160 bits).

Let's examine the transaction that was initiated by this address. Asset transfers require approved transactions that are also unreliable and unchangeable. They began by examining the digital signature technique and used it for that specific transaction. Data is encrypted and hashed using the digital signature. The secure hash is digitally signed before the receiver receives the actual data. The receiver can recompute the hash of the original data received and compare it to the received hash to verify the document's integrity [58]:

Step 1: Ascertain the transaction's data fields' hash.

Step 2: Involves using the participant's private key, which was used to start the transaction, to encrypt the hash. As a result, the transaction loses trustworthiness when it is digitally signed to approve it.

Step 3: The transaction was just recently updated with this hash. Others may confirm the transaction's hash beforehand by computing it and decrypting it with the sender's public key.

Compare the computed received hash with the digital signature after that. If the transaction matches, approve it. If not, it should be declined [56]. Figure 7 illustrates how the timestamp, nonce, account balances, and appropriateness of fees are all evaluated as part of a thorough transaction verification.

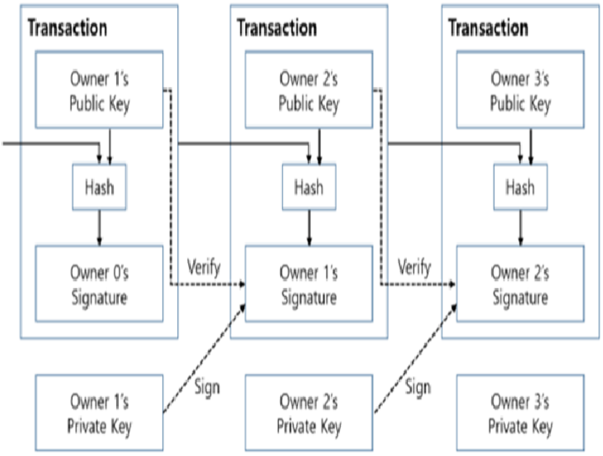

Figure 7. Operations Trust [49].

Blockchain, the distributed ledger technology supporting Bitcoin, holds potential value beyond its use as a digital currency, but its effectiveness relies on robust security. Establishing reliable initial conditions is important to ensure secure transactions [58]. The process is as follows: First, compute the hash value of the transaction's data fields. Next, encrypt this hash using the participant's private key to initiate the transaction. This encrypted hash is then added to the transaction as a digital signature. Others can verify the transaction by recalculating the hash and decrypting the digital signature using the sender's public key. Finally, compare the newly calculated hash with the decrypted hash from the signature; if they match, the transaction is approved. If they do not match, the transaction is rejected.

7.3. Chain Split

The Bitcoin protocol allows this chain to split into two chains during the subsequent cycle. It is doubtful that the following block in each chain will happen at the exact moment. Consequently, one of the chains is combined by the winner of the subsequent block generation cycle, which is then considered the accepted chain. The most recent block has been added to the main chain in this case. The longest and most dependable main chain is currently this one. The transactions from the other blocks are sent to the unconfirmed pool. The transactions from the other blocks are sent to the unconfirmed pool. Although there is a very slim probability that the main chain would split, the bitcoin protocol provides ways to rejoin it into a single chain within a cycle [58], [41].

Etherium accommodates more than one person we know by allowing them Runner-Up blocks and providing a little reward for these Runner-Up blocks. This incentive idea contributes to the chains' security. The Runner-Up chains do not get any additional blocks; only the leading chain does. In other words, when introduced, the runner-up blocks are preserved for six more blocks. It is possible to view a Blockchain with two blocks, one with a height of 4567 and the other with a height of 4557. The one introduced most recently is less trustworthy than the one further down the chain. Add the problem of double spending now. Digital money and other consumables might be one-time-use digital assets used again in transactions, either knowingly or unknowingly [42].

In a decentralized network like the Blockchain, there is no intermediary. We require a policy and an automated deterministic approach to address this problem. A rule for handling transactions and double-spending in Bitcoin is to accept the first transaction that refers to the digital asset and reject the subsequent transactions that do the same [42]. A combination of account numbers and a global nonce are used to overcome the double-spending problem in Ethereum. A global nonce is part of every transaction that an account starts. The nonce is raised after that [43]. The time stamp on the nonce in the transaction should be unique and verified to prevent duplicate uses of digital savings. Well-defined methods for handling exceptions increase the credibility of the Blockchain [44], [45].

A chain split in a blockchain happens when the network separates into two distinct chains due to differences in protocol rules or upgrades. This can occur through soft forks, backward-compatible updates, or rigid ones, creating permanent, incompatible chains. Chain splits have significant security implications. Soft forks can cause temporary consensus issues and reduced hash power, making the network more susceptible to 51% attacks. On the other hand, severe forks present more significant risks, including double spending through replay attacks and hash power dilution, which weakens both chains.

Additionally, hard forks can lead to community fragmentation and network instability. Chain splits can compromise a blockchain's security by creating vulnerabilities during the transition, reducing hash power, and weakening network consensus. To mitigate these risks and maintain the integrity of the Blockchain, it is essential to implement measures such as replay protection, consensus alignment, and user education [59], [60].

7.4. Forks

A fork in the road is where you should invest your trust. The words hard fork and soft fork, both types of forks, are most frequently employed in blockchain technology [60], [46]. According to reports, Ethereum hard split around block 4.7 million. On a high level, this section describes hard and soft forks. Forks are only a component of the organic evolution of nascent technology that makes Blockchain possible.

Regarding strength and trust, rigid and soft forks are at the forefront of managing extreme situations. The fork is a little break in the chain; on the Blockchain, such an event is viewed as normal. On the other hand, a slight process adjustment could occasionally be necessary, often by integrating fresh software into old procedures. This strategy was employed in the soft fork scenario to introduce the script concept to Bitcoin. This is comparable to a problem-solving software patch or bug fix. A significant modification to a protocol is referred to as a "hard fork." It's essential to keep in mind that the two chains that arise after a hard fork are incompatible. This was the case with the transition from Homestead to Metropolis Byzantium [47].

To fix a severe software flaw in a decentralized autonomous organization, the Ethereum protocol unexpectedly hard forked, splitting into Ethereum Classic and Ethereum Core Decentralized Autonomous Organization (DAO). A hard fork of Ethereum was anticipated. The Ethereum improvements included but not limited to (i) Suggestions for Ethereum Improvement (EIP and (ii) Transaction processing in parallel.

PoW consensus is still in use, except for every hundred blocks, and the PoS consensus protocol is used to assess it. The little reward for building blocks was also reduced from 5 ethers to 3 ethers. The terms "soft fork" and "hard fork" in the context of Blockchain relate, respectively, to the release of bug patches and new operating system versions. Forks are techniques that increase the robustness of the Blockchain system. Well-managed forks contribute to the growth of trust in the Blockchain by providing methods to handle unforeseen faults and planned improvements. You may find the documentation for other EIPs here [54].

Forks, whether hard or soft, can lead to a loss of trust in the blockchain network if users perceive instability, conflicts within the community, or concerns about future splits. Trust is critical for maintaining a solid user base and secure transaction processing. A loss of trust can result in reduced network participation and value. Furthermore, forks can provide new attack vectors for malicious actors. For example, they may exploit the uncertainty or confusion following a fork to launch phishing attacks, impersonate network validators, or attempt to disrupt exchanges and services handling both versions of the Blockchain. Forks, whether hard or soft, come with built-in security risks that can undermine blockchain agreements, divide communities, and create new vulnerabilities. Hard forks risk creating permanent network divisions, spreading resources thin, and enabling replay attacks, while soft forks can temporarily weaken agreement, introduce bugs in new rules, and pose centralization risks. To address these risks, clear communication, broad community support, robust replay protection, and comprehensive testing are needed before making any changes to the protocol. In both scenarios, the blockchain ecosystem needs to balance the need for innovation and upgrades and the necessity of upholding network security and integrity [67].

8. Security Improvement

Another method of demonstrating security is by using the encryption algorithm. As seen below, several methods are frequently used for chain security as well as effective validation and verification [61], [62]:

8.1. Asymmetric Key Encryption

Using two public and private keys that are mathematically related but not identical, asymmetric cryptography encrypts data [61]. While decryption uses the private key, encryption uses the public key. Each key has a particular function, unlike symmetric vital approaches that utilize the same key for encryption and decryption.

It should be noted that it is computationally impossible to deduce the private key from the public key. As a result, public keys may be freely distributed, enabling users to easily and quickly encrypt data and validate digital signatures. Private keys can be kept secret, ensuring that only those who possess them can establish digital signatures and decode content [61], [62] figure 8. The Keys' Mechanism (PoW).

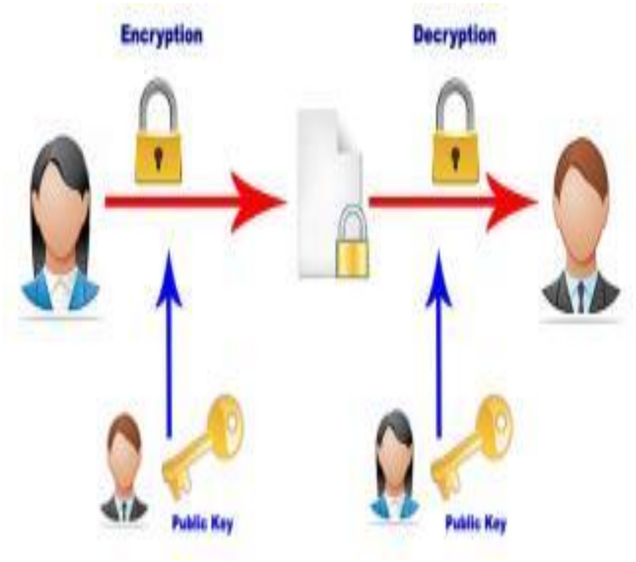

Figure 8. Keys Mechanism (Public and Private) [61].

Public keys are stored in digital certificates to facilitate secure distribution and transmission, as their size makes them impractical to remember. In contrast, private keys are not shared; they are securely stored within the software or operating system you use or on specialized hardware (like a USB token or hardware security module) equipped with drivers that ensure compatibility with your software or system. These digital certificates are issued by Certificate Authorities (CAs) [62], [63].

8.2. Public-Key Cryptography

Anyone may use the public key to encrypt data because it is available. On the other hand, the private key is kept a secret and can only be used by those with it to decode data [61]. Suppose that Donald Trump posts a private message on Facebook. Facebook must ensure that nobody in the middle (like the NSA or an internet service provider) can see the communication when the former president sends it. Faster transaction verification through network communication is one benefit of this encryption. There are several key benefits to using public and private key encryption. First, the public key is accessible to everyone, allowing open communication without compromising security. Second, the private key must remain secure; if exposed, intermediaries or unauthorized users could potentially decode sensitive information. This public-private key pairing lets computers quickly encrypt and decrypt messages, streamlining secure communication. Furthermore, decrypting an encrypted message without the private key is extremely difficult and time-consuming, potentially taking millions of years to break.

The kernel of the situation is that each public-key cryptography technique has a unique trapdoor function. A trapdoor function has just one way to be calculated, or at least one that can be quickly calculated (with current computers, in fewer than millions of years) [64]. There is no function for a trapdoor:

$$A + B = C \tag{1}$$

If the private key is one of the two main components of the public key, the public key is a huge number. This is an excellent example of a trapdoor function since adding the numbers in the private key together is straightforward to obtain the public key. Recreating the private key using a computer will take much time, even if you have the public key [64].

8.3. Rivest Shamir Adleman (RSA)

The RSA protocol supports public-key encryption and digital signatures [65]. This approach lets the receiver transmit a message that can be encoded by any sender but can only be decoded by the recipient. As a result, anybody can encrypt and deliver data over any public channel; only the first receiver can decode the data [66].

The user can transfer the encrypted data after completing authentication RSA. The user's private key will be used to encrypt the data. Next, the encrypted data and the public key will be delivered. On the other hand, the proxy will get both the public key and the encrypted data, and it will use its private key to decode it [66].

8.4. Elliptic Curve Cryptography (ECC)

The key pair in the Bitcoin and Ethereum blockchains are generated using an algorithm from the ECC family [65]. as seen in figure 9.

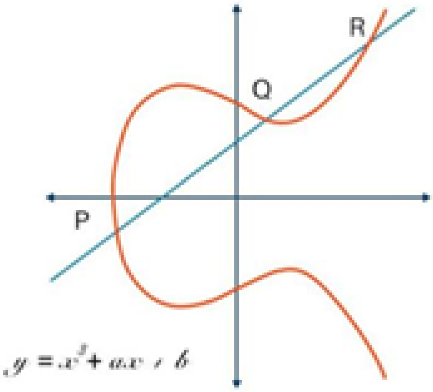

Figure 9. ECC Carve [67].

The private key must be 200 digits or more to be considered safe in actual cryptography. You would use ECC for the same reasons RSA is used. RSA and ECC generate a public and private key that permits secure communication between two parties. A 256-bit key in ECC offers almost the same amount of security as a 3072-bit key in RSA, which is one advantage of ECC. ECC permits 10% of RSA's storage and bandwidth to be used by devices with constrained resources, such cellphones, embedded computers, and cryptocurrency networks. Figure 10 illustrates how both bitcoin and Ethereum use ECC-based algorithms for encryption.

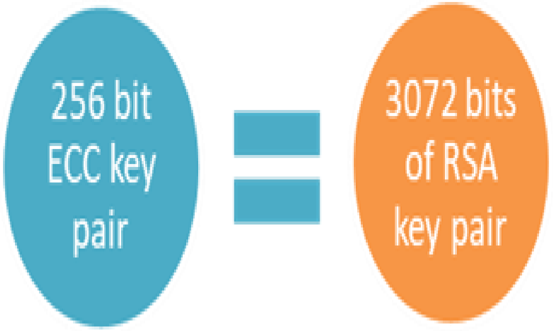

Figure 10. Comparing key pair of ECC and RSA [67].

Encryption methods are vital for safeguarding the security and integrity of blockchain systems. The following are some fundamental encryption techniques that play a crucial role in securing blockchain operations [68]:

8.4.1. Multi-Factor Authentication (MFA)

MFA is a robust security measure that requires users to verify their identity using a combination of different credentials before being granted access to blockchain systems. These credentials can include traditional elements like passwords or PINs and more advanced factors such as one-time passwords (OTPs), security tokens, smartphones, and biometric data like fingerprints or facial recognition. In the context of Blockchain, MFA plays a critical role in fortifying the security of wallets and accounts by introducing an additional layer of protection beyond just passwords. This means that even if a password is compromised, the use of supplementary verification methods helps to thwart unauthorized access to valuable blockchain assets, thereby mitigating the risks associated with hacking and theft [68].

8.4.2. Secure Key Management

Private keys are critical in blockchain technology, enabling users to access assets, sign transactions, and secure communications. Employing robust key management methods is essential to ensure these keys are stored, transmitted, and managed securely [68].

Hardware Security Modules (HSMs) are physical devices designed to securely generate, store, and protect private keys, reducing the risk of exposure to malware and external threats. Cold Storage, which involves storing private keys offline (in cold wallets), protects against online attacks. Examples of cold storage solutions include hardware wallets and paper wallets, both of which remain disconnected from the internet.

Hierarchical Deterministic (HD) Wallets can generate multiple private keys from a single seed, simplifying the backup and recovery process while maintaining a high level of security. Implementing secure essential management practices is crucial for blockchain users and organizations to retain control over their assets, effectively minimizing the risks of critical loss or unauthorized access.

8.4.3. Smart Contract Auditing

Smart contracts are essentially self-executing programs that exist and operate on blockchains. They are commonly utilized in decentralized applications (dApps), token issuance, and automated processes. However, it's essential to be aware that bugs or vulnerabilities within intelligent contracts can potentially result in exploits or loss of funds. To mitigate such risks, security experts conduct smart contract auditing before deployment. This process involves a comprehensive examination of the code to pinpoint any potential flaws, logic errors, or vulnerabilities. Specifically, auditors scrutinize for issues such as reentrancy attacks, which have the potential to facilitate double withdrawals or other malicious activities. Moreover, formal verification techniques employing mathematical methods prove that a contract behaves as expected under various scenarios. Ultimately, regular smart contract audits play a critical role in upholding trust in blockchain applications, minimizing risk, and averting costly exploits, as evidenced by past incidents involving platforms like The DAO or Poly Network [68].

By leveraging advanced security measures such as MFA, Secure Key Management, and Smart Contract Auditing, blockchain systems can bolster their defenses against unauthorized access and attacks. These encryption-based strategies play a crucial role in safeguarding user assets, mitigating potential vulnerabilities, and instilling confidence in the reliability and robustness of blockchain platforms.

9. Cryptographic Considered Secure

A particular kind of hash function that works well for cryptography is called a cryptographic hash function [27]. A cryptographic hash function has to have the qualities listed and explained below [69] to be thought of as secure:

9.1. Deterministic

You will always get the same output no matter how often you run a specific input through a hash function. Determining is crucial since using different hashes each time will make it difficult to follow the input.

9.2. Quick Computation

The hash function should quickly return the input hash. The system will be ineffective if the process is not swift enough.

9.3. Pre-Image Resistance

It only requires comparing the hashes of all numbers from 1 to 6, which is simple. You can compare hashes and return the original data since hash functions are deterministic and produce the same hash for any input. This, however, only functions when there is significantly less data available. The primary strategy for finding the original input is known as the "brute-force approach." In the brute-force approach, random data is collected hashed, the result is compared to the target hash, and the process is repeated until a match is discovered.

9.4. Changes in Input Hash

It is a critical function since hashing contributes to immutability, one of the Blockchain's most outstanding features. The hash will change significantly even if users make a minor change to the input. Using SHA-256, let's test it out: Check out how much the output hash has changed after the case of the initial letter of the input has been altered.

9.5. Collision Resistant

With two different inputs, A and B, where H(A) and H(B) are their respective hashes, H(A) can't equal H(B) (B). A unique hash will be assigned to each input. The square root of N random items colliding has a 50% chance of happening when there are N possible outcomes for an event.

Consider a 128-bit hash that has two 128 potential results. In other words, it is far simpler to overcome impact resistance than pre-image resistance. Even though no hash function is collision-free, identifying a collision often takes a lot of time. When employing a function like SHA-256, it is acceptable to assume that:

$$A = B \qquad \text{if} \quad H(A) = H(B) \tag{2}$$

9.6. Puzzle Friendly

The consequences and effects of just this one trait for Bitcoin are enormous. It is a fascinating property Puzzle Friendly. If k is chosen from a distribution with maximum, it is impossible to find an input x such that H(k|x) = Y for every output "Y." That most likely means that the distribution from which the value is drawn is very scattered; nonetheless, there is very little likelihood of selecting a random value. An assortment with a low min-entropy is one between 1 and 5. Indicating concatenation is the symbol "|." Concatenation is the procedure used to connect two strings. For, Let's say your output value is "Y." If you choose a random value "k" from a wide distribution, it is impossible to find a number X such that the hash of the concatenation of k and x yields the result Y. It's essential to remember that "infeasible" does not equate to "impossible," as individuals often achieve this.

10. Scalability with Blockchain-Cloud and Its Impact on Security

In cloud computing, scalability enables service providers to adjust resources to meet changing user demands. However, this flexibility can pose security risks, as over- or under-provisioning may lead to data breaches or denial-of-service (DoS) attacks. Managing security in distributed environments is complex, as organizations must secure a larger potential attack surface. As cloud systems scale, controlling identity and access becomes increasingly challenging, raising concerns about unauthorized access and data leaks [20].

Cloud computing faces critical vulnerabilities such as DDoS attacks and data breaches. DDoS attacks inundate cloud services with excessive traffic, causing downtime and increased operational costs. Cloud providers counter these attacks with traffic filtering and DDoS protection services. Data breaches result from misconfigurations, weak access controls, or insider threats, leading to significant financial and reputational damage. Encryption, robust identity management, and continuous monitoring are crucial for preventing these vulnerabilities and ensuring cloud security.

Blockchain's decentralized and consensus-based nature creates scalability challenges. As more nodes join and transaction volume increases, the network may experience slower processing times and higher costs. Consensus mechanisms like PoW and PoS become bottlenecks as the network scales, and 51% of attacks become more feasible with large, centralized mining pools or participants controlling most of the computational power.

10.1. Interplay Between Scalability and Security in Both Systems

The challenge of scalability in Blockchain and Cloud Computing is multifaceted, as it involves the delicate balance of maintaining security while expanding capacity. In the realm of Cloud Computing, rapid scalability can significantly complicate the task of securing distributed resources, thereby making it increasingly challenging to enforce robust security controls. Similarly, in the context of Blockchain, endeavors to enhance scalability through methods like sharding or layer-two solutions (such as the Lightning Network for Bitcoin) can potentially introduce new vulnerabilities. This is because these methods may reduce redundancy and decentralization, vital for ensuring security within the Blockchain ecosystem [70], [71].

10.2. Addressing Scalability with Blockchain-Cloud Integration

Integrating Blockchain's decentralized security model with the scalability benefits of cloud computing poses a significant challenge for both Blockchain and Cloud Computing infrastructures. It is crucial to address this challenge to ensure that these systems can effectively accommodate the growing demands of users while upholding their performance and security standards [68], [71].

10.3. Scalability in Cloud Computing and Its Security Implications

Cloud computing is designed to be highly scalable, allowing for the seamless adjustment of resources such as storage, processing power, and network bandwidth based on demand. However, cloud infrastructures expand to accommodate a growing user base and a more comprehensive range of applications [71].

As cloud systems expand, the number of potential attack entry points increases, making securing all nodes and services harder. Improper resource allocation during scaling, such as over-provisioning or under-provisioning, can lead to vulnerabilities. For example, under-provisioning can cause performance issues and make systems vulnerable to DoS attacks, while over-provisioning can increase unnecessary costs and complicate access control management. Additionally, managing and securing large volumes of data across distributed locations becomes more complex as cloud services scale, increasing the risk of data breaches and loss, especially in multi-tenant environments where multiple clients share infrastructure.

10.4. Scalability in Blockchain and Its Security Implications

Blockchain technology faces unique scalability challenges due to its decentralized and consensus-based nature, where multiple nodes must verify each transaction. This requirement can lead to slower processing times and increased operational costs as the network expands, posing security implications that affect its ability to process transactions efficiently and securely [71]. One major issue is the limitations of consensus mechanisms: blockchains that use protocols like PoW or PoS struggle with increased network activity, leading to slower transaction speeds and higher energy consumption. These limitations can create bottlenecks and even leave the system susceptible to 51% attacks, where an entity controlling most of the network's computational power could potentially alter transaction records. Another issue is the risk of centralization; some blockchain networks attempt to enhance scalability by implementing centralized solutions or reducing the number of nodes required for transaction validation. While this improves speed, it weakens the security provided by decentralization, introducing single points of failure and the risk of collusion among influential participants. Finally, delays in transaction validation are common in busy networks, as the time needed to validate and confirm transactions increases with higher network traffic, causing transaction delays.

10.5. Addressing Scalability with Blockchain-Cloud Integration

Combining Blockchain and Cloud Computing technologies can effectively address scalability challenges while maintaining robust security measures [71]. One potential approach is the implementation of layered solutions in Blockchain, such as the Lightning Network for Bitcoin. These innovative technologies are designed to reduce the main Blockchain's load by diverting transaction processing to secondary layers, enabling the network to handle a higher volume of transactions without overburdening the consensus mechanism. This enhances the Blockchain ecosystem's scalability while ensuring the underlying infrastructure remains secure and robust. Another technique is sharding, which involves dividing the Blockchain network into smaller, more manageable segments called shards. Each shard handles a specific portion of transactions, aiming to improve the network's scalability. However, sharding can introduce security challenges, as each shard may experience reduced decentralization, making it more susceptible to security breaches and attacks. Additionally, integrating blockchain technology into cloud computing platforms for decentralized resource management presents a promising solution. This decentralized cloud approach allows platforms to allocate resources securely and efficiently dynamically, leveraging Blockchain's transparency and trust features. Furthermore, blockchain technology ensures data integrity and provides tamper-proof auditing within cloud environments, enhancing scalability and security in cloud-based systems.

10.6. Balancing Scalability and Security

The expansion of Blockchain and Cloud Computing must be carefully managed to guarantee that security is not threatened. Measures to improve scalability should not undermine decentralization, integrity, or confidentiality. Both cloud service providers and Blockchain networks should seek solutions that balance performance, cost, and security, ensuring that the scalability enhancements are implemented without sacrificing crucial aspects of the systems. They were delving into alternative consensus mechanisms, such as PoA or DPoS, which offer enhanced energy efficiency and scalability while upholding robust security measures. Leveraging hybrid models is essential for optimizing transaction processing. By employing these models, non-sensitive transactions can be processed off-chain (in the case of Blockchain) or off-cloud (in the case of Cloud Computing). This strategy effectively alleviates congestion and significantly enhances scalability [71].

10.7. Here Are Real-World Studies Involving Blockchain and Cloud Computing

This section provides summaries of notable security incidents related to blockchain and cloud computing, as detailed in table 1. These incidents highlight vulnerabilities stemming from misconfigurations, inadequate access controls, and weaknesses in smart contract security. Each case underscores critical lessons for maintaining robust cloud security, ensuring proper configuration management, and conducting comprehensive code audits to prevent similar breaches in the future.

Table 1. Summary of significant security incidents related to blockchain and cloud computing

Incident	Description	Key Lesson
Capital One Data Breach (2019)	Misconfiguration of an AWS firewall exposed sensitive information of over 100 million customers, leading to an $80 million fine.	Highlights the importance of proper cloud configuration management.
Ethereum DAO Hack (2016)	A vulnerability in the DAO's smart contract allowed an attacker to steal $60 million in Ether, prompting a hard fork to recover the funds.	Stresses the need for comprehensive code audits.
Microsoft Azure Data Breach (2020)	A misconfigured database on the Azure cloud platform exposed sensitive customer data, raising concerns about cloud data security.	Emphasizes the importance of regular configuration reviews.
Bitfinex Hack (2016)	The compromise of multi-signature wallets led to the unauthorized transfer of $72 million in Bitcoin.	Reinforces the need for robust security measures in cryptocurrency exchanges.
Parity Wallet Hack (2017)	The breach in multi-signature wallets allowed a hacker to steal over $30 million in Ether.	Underlines the necessity of thorough security audits for smart contracts.
AWS S3 Bucket Exposures	Organizations like Verizon and Accellion suffered data leaks due to publicly accessible, misconfigured S3 buckets.	Highlights the need for stringent access controls in cloud storage settings.

Capital One Data Breach (2019): The misconfiguration of an AWS firewall resulted in the exposure of sensitive information belonging to over 100 million customers. This incident led to Capital One being fined $80 million, a stark reminder of the critical importance of proper cloud configuration management.

Ethereum DAO Hack (2016): In the past, a vulnerability in the DAO's smart contract resulted in an attacker stealing $60 million worth of Ether. This unfortunate incident prompted a contentious hard fork to recover the funds, a powerful reminder of the critical importance of conducting comprehensive code audits.

Microsoft Azure Data Breach (2020): A misconfigured database in the Azure cloud platform exposed sensitive data from its customers. This incident has sparked concerns regarding cloud data security and emphasized the importance of conducting regular configuration reviews to prevent such breaches.

Bitfinex Hack (2016): Bitfinex's multi-signature wallets were compromised, leading to the unauthorized transfer of $72 million in Bitcoin. This significant breach emphasized the critical need for implementing and maintaining strong security measures within cryptocurrency exchanges.

Parity Wallet Hack (2017): The breach in Parity's multi-signature wallets resulted in a hacker absconding with over $30 million worth of Ether. This incident underscores the importance of conducting comprehensive security audits while developing smart contracts.

AWS S3 Bucket Exposures: Numerous organizations, such as Verizon and Accellion, experienced data leaks due to misconfigured S3 buckets left publicly accessible, highlighting the importance of implementing stringent access controls.

The recent security breaches are a stark reminder of the formidable hurdles confronting organizations leveraging Blockchain and Cloud Computing technologies. These incidents underscore the critical importance of implementing robust security measures, conducting thorough audits, and adhering to best practices in development and configuration. These actions are essential for fortifying defenses against vulnerabilities and minimizing the potential fallout from security breaches.

The vulnerabilities and challenges faced by different consensus mechanisms in blockchain technology are crucial factors to consider in ensuring a network's security, efficiency, and decentralization. In PoW networks, the risk of 51% attacks allows a single entity with the majority computational power to manipulate transactions. Similarly, wealthier or more influential participants can dominate the network, posing similar threats in PoS and DPoS systems. Consensus mechanisms like PoW and BFT may encounter scalability challenges due to their reliance on significant computational resources and node communication, leading to limited transaction throughput. Despite being designed for decentralization, consensus models like PoS, DPoS, and PoA introduce centralization risks. Wealth concentration, voting mechanisms, and reliance on trusted validators can result in unequal power distribution among participants. While PoW offers robust security, it is energy-intensive.

On the other hand, PoS and DPoS are more energy-efficient but may face significant security trade-offs as network centralization increases. Choosing a trust mechanism significantly impacts a blockchain's security, efficiency, and decentralization. It is essential to carefully consider the trade-offs associated with each consensus model based on the specific needs of the network and its intended use cases.

11. Conclusion

The Cloud Computing environment should allow for universal access to data storage services. However, there are a variety of security issues explained carefully in this article that might prevent this from happening. These features, supported by an attractive and economical approach, have led to growing support for this article. The primary reason for bitcoin transactions is the exchange of bitcoins. The text strings corresponding to the bitcoins being traded are unique to each transaction.

Meanwhile, some Blockchain systems record the ownership of the assets or shares that are a part of a transaction. A private key, a long number generated by an algorithm designed to produce a random and unique result and linked to any data in the bitcoin system, identifies ownership. The second level will be availability, as providers can be victims of attacks that stop their operations running. The attacks are the consequence of preserving private critical vulnerabilities; these assaults are not the result of a weakness in bitcoin's security. All sensitive and essential data must be encrypted before posting to prevent users from accessing each other's files. This paper discusses security issues that arise with the user to prevent users from accessing each other's files; all sensitive and essential data must be encrypted before being posted. There are no controls to regulate access to sensitive files, including those containing passwords. The data transfer security within the cloud is currently not ensured by using file encryption techniques.

Moreover, both Blockchain and Cloud Computing face significant challenges related to scalability. Overcoming these challenges requires a thorough understanding of the security implications involved. The future success of both systems must embrace innovative technologies and hybrid solutions that enhance scalability without compromising security. These incidents underscore the urgent need for enhanced security measures and best practices in Blockchain and Cloud Computing environments.

12. Declarations

12.1. Author Contributions

Conceptualization: H.A.A., M.A.; Methodology: M.A.; Software: H.A.A.; Validation: H.A.A., M.A.; Formal Analysis: H.A.A., M.A.; Investigation: H.A.A.; Resources: M.A.; Data Curation: M.A.; Writing—Original Draft Preparation: H.A.A., M.A.; Writing—Review and Editing: M.A., H.A.A.; Visualization: H.A.A.; All authors have read and agreed to the published version of the manuscript.

12.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

12.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

12.4. Institutional Review Board Statement

Not applicable.

12.5. Informed Consent Statement

Not applicable.

12.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] H. A. Albaroodi, M. Abomaali, and S. Manickam, *Iraqi's Organizations Awareness to Prompt Open Source Cloud Computing (OSCC) in Their Service: A Study*, vol. 1132 CCIS, no.1, pp. 305-319. 2020. doi: 10.1007/978-981-15-2693-0_22.

[2] H. A. Albaroodi, M. Abomaali, A. Ismael, and S. Manickam, "Effectiveness of Open Source Cloud Computing in Developing Countries: Empirical Study," *ARPN J. Eng. Appl. Sci.*, vol. 14, no. 4-Supplement1, pp. 7313-7319, 2019, doi: 10.36478/JEASCI.2019.7313.7319.

[3] H. Albaroodi, S. Manickam, and P. Singh, "Critical review of openstack security: Issues and weaknesses," *J. Comput. Sci.*, vol. 10, no. 1, pp. 23-33, 2013, doi: 10.3844/jcssp.2014.23.33.

[4] W. F. Silvano and R. Marcelino, "Iota Tangle: A cryptocurrency to communicate Internet-of-Things data," *Futur. Gener. Comput. Syst.*, vol. 112, no. 11 ,pp. 307–319, 2020.

[5] S. J. Pee, J. H. Nans, and J. W. Jans, "A simple blockchain-based peer-to-peer water trading system leveraging smart contracts," in *Proceedings on the International Conference on Internet Computing (ICOMP)*, The Steering Committee of The World Congress in Computer Science, Computer, vol. 2018, no. 3, pp. 63–68, 2018.

[6] I. Zikratov, A. Kuzmin, V. Akimenko, V. Niculichev, and L. Yalansky, "Ensuring data integrity using blockchain technology," *20th Conference of Open Innovations Association (FRUCT)*, IEEE, vol. 2017, no. 4, pp. 534–539, 2017.

[7] H. Albaroodi, S. Manickam, and M. Anbar, "A proposed framework for outsourcing and secure encrypted data on OpenStack object storage (Swift)," *J. Comput. Sci.*, vol. 11, no. 3, pp. 590.597 , 2015, doi: 10.3844/jcssp.2015.590.597.

[8] H. Lo, R. Wang, and J. P. Garbini, "The state of enterprise software 2009," *Forrester Res. Cambridge*, vol. 2009, no. 6, pp.11, 2009.

[9] L. Yue, H. Junqin, Q. Shengzhi, and W. Ruijin, "Big data model of security sharing based on blockchain," in *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*, IEEE, vol. 2017, no. 8 , pp. 117–121, 2017.

[10] Dannen, C., ," Introducing Ethereum and solidity", *Berkeley: Apress,.Springer*, vol. 1, no. 1, pp. 159-160, 2017.

[11] G. J. Holzmann, "An analysis of bitstate hashing," Form. methods Syst. Des., vol. 13, no. 3, pp. 289–307, 1998.

[12] P. Kasireddy, "How does Ethereum work, anyway," Medium, vol. 2017, no. 7, pp 219–266, 2017.

[13] M. Fleder, M. S. Kester, and S. Pillai, "Bitcoin transaction graph analysis," *arXiv Prepr. arXiv1502.01657*, vol. 2015, no. 2, pp. 1-8, 2015.

[14] M. Szydlo, "Merkle tree traversal in log space and time," in *International Conference on the Theory and Applications of Cryptographic Techniques*," Springer, vol. 2004, no. 5, pp. 541–554, 2004.

[15] T. Skopal, J. Pokorný, M. Krátký, and V. Snášel, "Revisiting M-tree building principles," in *East European Conference on Advances in Databases and Information Systems*, Springer, vol. 2003, no. 9, pp. 148–162, 2003.

[16] R. Qin, Y. Yuan, and F.-Y. Wang, "Research on the selection strategies of blockchain mining pools," *IEEE Trans. Comput. Soc. Syst.*, vol. 5, no. 3, pp. 748–757, 2018.

[17] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1026–1037, 2004.

[18] Kroll JA, Davey IC, Felten EW, "The economics of Bitcoin mining", or Bitcoin in the presence of adversaries. InProceedings of WEIS Jun 11 2013 , vol. 2013, No. 11, pp. 1- 21, 2013.

[19] M. Jakobsson, T. Leighton, S. Micali, and M. Szydlo, "Fractal Merkle tree representation and traversal," in *Cryptographers' Track at the RSA Conference*, Springer, pp. 314–326, 2003.

[20] Chakraborty, Trisha, Shaswata Mitra, Sudip Mittal, and Maxwell Young. "A policy driven AI-assisted pow framework." In 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental, IEEE, Volume (DSN-S), vol. 2022, no. 6, pp. 37-38, 2022.

[21] N. Kheshaifaty and A. Gutub, "Preventing multiple accessing attacks via efficient integration of captcha crypto hash functions," *Int. J. Comput. Sci. Netw. Secur.*, vol. 20, no. 9, pp. 16–28, 2020.

[22] R. Cross, J. Liedtka, and L. Weiss, "A practical guide to social networks," *Harv. Bus. Rev.*, vol. 83, no. 3, pp. 124–132, 2005.

[23] A. Hooper and D. Holtbrügge, "Blockchain technology in international business: changing the agenda for global governance," *Rev. Int. Bus. Strateg.*, vol. 30, no. 2, pp. 183–200, 2020.

[24] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, vol. 2016, no. 10, pp. 3–16, 2016.

[25] H. A. Albaroodi and M. Anbar, "ETHEREUM-INSPIRED ACCESS MANAGEMENT ACCOUNT CONTROL FOR A SECURED DECENTRALIZED CLOUD STORAGE," *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 7, pp. 2229 - 2242, 2022.

[26] De Kruijff, Joost, and Hans Weigand , "Towards a blockchain ontology," *Netherlands, pdfs. Semant*, vol. 2017, no. 10, pp. 3–16, 2017.

[27] M. Kuhn, F. Funk, G. Zhang, and J. Franke, "Blockchain-based application for the traceability of complex assembly structures," *J. Manuf. Syst.*, vol. 59, no. 4, pp. 617–630, 2021.

[28] K. Saini, "A future's dominant technology blockchain: Digital transformation," in *2018 international conference on computing, power and communication technologies (GUCON)*, IEEE, vol. 2018, no. 3, pp. 937–940, 2018.

[29] I. Shilov and D. Zakoldaev, "Multidimensional Blockchain as Robust Distributed Ledger," in *2022 31st Conference of Open Innovations Association (FRUCT)*, IEEE, vol. 2022, no. 5, pp. 313–319, 2022.

[30] N. El Ioini and C. Pahl, "A review of distributed ledger technologies," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, Springer, vol. 2018, no. 10, pp. 277–288, 2018.

[31] M. A. Khan, F. Algarni, and M. T. Quasim, "Decentralised internet of things," in *Decentralised Internet of Things*, Springer, vol. 2020, no. 2, pp. 3–20, 2020.

[32] B. Preneel, "Cryptographic hash functions," *Eur. Trans. Telecommun.*, vol. 5, no. 4, pp. 431–448, 1994.

[33] D. R. Stinson, "Some observations on the theory of cryptographic hash functions," *Des. Codes Cryptogr.*, vol. 38, no. 2, pp. 259–277, 2006.

[34] M. OA and B. AS, "SIMULATION OF THE RAINBOW ATTACK ON THE SHA-256 HASH FUNCTION," *J. Theor. Appl. Inf. Technol.*, vol. 101, no. 4, pp. 1594-1603, 2023.

[35] Landau, Susan. "Find me a hash." *Notices of the AMS* 53,vol. 53, no. 3, pp. 330- 332, 2006

[36] 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, 2017.

[37] F. A. Khan, M. Asif, A. Ahmad, M. Alharbi, and H. Aljuaid, "Blockchain technology, improvement suggestions, security challenges on smart grid and its application in healthcare for sustainable development," *Sustain. Cities Soc.*, vol. 55, no. 4, pp. 102018-?, 2020.

[38] Albaroodi, Hala A., and Mohammed Anbar. "Saba: Secure Approach Based On Anomaly And Signature-Based Detection Mechanism For Detecting Abnormal Activities In Blockchain Network." *Journal Of Theoretical And Applied Information Technology*, vol. 101, no. 17, pp. 6885-6896, 2023.

[39] C. Gupta and A. Mahajan, "Evaluation of proof-of-work consensus algorithm for blockchain networks," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, vol. 2020, no. 7 pp. 1–7, 2020.

[40] E. Ephrati and J. S. Rosenschein, "The Clarke Tax as a Consensus Mechanism Among Automated Agents.," in *AAAI*, vol. 1, no. 7 , pp. 173–178, 1991.

[41] C. Dierksmeier and P. Seele, "Cryptocurrencies and business ethics," *J. Bus. Ethics*, vol. 152, no. 1, pp. 1–14, 2018.

[42] Poelstra, Andrew. "Distributed consensus from proof of stake is impossible." *Self-published Paper* (2014), vol. 2014, no. 5. Pp.1-5, 2014.

[43] Laurie, Ben, and Richard Clayton. "Proof-of-work proves not to work; version 0.2." In *Workshop on economics and information, security*. 2004., vol. 2004, no. 9, pp.1-11, 2004.

[44] F. Leal, A. E. Chis, and H. González–Vélez, "Performance evaluation of private ethereum networks," *SN Comput. Sci.*, vol. 1, no. 5, pp. 1–17, 2020.

[45] D. Vujičić, D. Jagodić, and S. Ranđić, "Blockchain technology, bitcoin, and Ethereum: A brief overview," in *2018 17th international symposium infoteh-jahorina (infoteh)*, IEEE, vol. 2018, no. 3, pp. 1–6, 2018.

[46] Q. Luo, J. Zhou, S. Yu, and D. Xiao, "Stereo matching and occlusion detection with integrity and illusion sensitivity," *Pattern Recognit. Lett.*, vol. 24, no. 9–10, pp. 1143–1149, 2003.

[47] C. Wang, X. Chu, and Y. Qin, "Measurement and analysis of the bitcoin networks: A view from mining pools," in *2020 6th International Conference on Big Data Computing and Communications (BIGCOM)*, IEEE, vol. 2020, no. 7, pp. 180–188, 2020.

[48] T. D. Perez and S. Pagliarini, "A survey on split manufacturing: Attacks, defenses, and challenges," *IEEE Access*, vol. 8, no. 10, pp. 184013–184035, 2020.

[49] A. Begum, A. Tareq, M. Sultana, M. Sohel, T. Rahman, and A. Sarwar, "Blockchain attacks analysis and a model to solve double spending attack," *Int. J. Mach. Learn. Comput.*, vol. 10, no. 2, pp. 352–357, 2020.

[50] R. Bhat, M. R. Ansari, and R. Khanam, "Effect of integrated power and clock networks on combinational circuits," *Int. J. Reconfigurable Embed. Syst.*, vol. 9, no. 3, pp. 242 – 248 , 2020.

[51] J. Ho *et al.*, "Can digital pathology result in cost savings? A financial projection for digital pathology implementation at a large integrated health care organization," *J. Pathol. Inform.*, vol. 5, no. 1, pp. 1-33, 2014.

[52] A. Zamyatin, N. Stifter, A. Judmayer, P. Schindler, E. Weippl, and W. J. Knottenbelt, "A wild velvet fork appears! inclusive blockchain protocol changes in practice," in *International Conference on Financial Cryptography and Data Security*, Springer, vol. 2018, no. 2, pp. 31–42, 2018.

[53] S. Bhatia and A. D. Wright de Hernandez, "Blockchain is already here. What does that mean for records management and archives?," *J. Arch. Organ.*, vol. 16, no. 1, pp. 75–84, 2019.

[54] B. Cao *et al.*, "Performance analysis and comparison of PoW, PoS and DAG based blockchains," *Digit. Commun. Networks*, vol. 6, no. 4, pp. 480–485, 2020.

[55] E. Toufaily and T. Zalan, "In Blockchain we trust? Demystifying the 'trust' mechanism in blockchain ecosystems," *Technol. Forecast. Soc. Change*, vol. 206 , no. 9, pp. 123574 , 2024.

[56] S. Agarwal, G. Atondo-Siu, M. Ordekian, A. Hutchings, E. Mariconti, and M. Vasek, "Short Paper: DeFi Deception—Uncovering the Prevalence of Rugpulls in Cryptocurrency Projects," in *International Conference on Financial Cryptography and Data Security*, Springer, vol.2023, no. 12, pp. 363–372, 2023.

[57] Bastiaan, Martijn. "Preventing the 51%-attack: a stochastic analysis of two phase proof of work in bitcoin (2015)." *pp.1-10,2015.* Link : https://fmt.ewi.utwente.nl/media/175.pdf, vol. 2015, no. 1, pp. 1-10, 2015.

[58] F. A. Aponte-Novoa, A. L. S. Orozco, R. Villanueva-Polanco, and P. Wightman, "The 51% attack on blockchains: A mining behavior study," *IEEE Access*, vol. 9, pp. 140549–140564, 2021.

[59] M. Šipek, M. Žagar, N. Drašković, and B. Mihaljević, "Blockchain as an IoT intermediary," in *Interactive Mobile Communication, Technologies and Learning*, Springer, vol. 2021, no. 10, pp. 423–430, 2021.

[60] C. Li, R. Xu, and L. Duan, "Characterizing Coin-Based Voting Governance in DPoS Blockchains," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 17, no.1, pp. 1148–1152, 2023.

[61] P.-L. Aublin, S. Ben Mokhtar, and V. Quéma, "Rbft: Redundant byzantine fault tolerance," in *2013 IEEE 33rd international conference on distributed computing systems*, IEEE, vol. 2013, no. 7 , pp. 297–306, 2013.

[62] R. Ramadoss, "Blockchain technology: An overview," *IEEE Potentials*, vol. 41, no. 6, pp. 6–12, 2022.

[63] M. A. Manolache, S. Manolache, and N. Tapus, "Decision making using the blockchain proof of authority consensus," *Procedia Comput. Sci.*, vol. 199, no. 7, pp. 580–588, 2022.

[64] E. Aarti, "A Review of Blockchain Technology," *Smart City Infrastruct. Blockchain Perspect.*, vol. 2022, no. 2, pp. 225–246, 2022.

[65] J.Han, "Chain-split evaluation in deductive databases," *IEEE Trans. Knowl. Data Eng.*, vol. 7, no. 2, pp. 261–273, 1995.

[66] D. Luo, Q. Cai, G. Sun, H. Yu, and D. Niyato, "Split-Chain based Efficient Blockchain-Assisted Cross-Domain Authentication for IoT," *IEEE Trans. Netw. Serv. Manag.*,vol. 2024, no. 3, pp.3209 - 3223, 2024.

[67] S. Zhou, B. Vasilescu, and C. Kästner, "How has forking changed in the last 20 years? a study of hard forks on github," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, vol.2020, no. 10, pp. 445–456, 2020.

[68] D. Pointcheval, "Asymmetric cryptography and practical security," *J. Telecommun. Inf. Technol.*, vol. 2002, no. 4, pp. 41–56, 2002.

[69] B. Alaya and L. Sellami, "Clustering method and symmetric/asymmetric cryptography scheme adapted to securing urban VANET networks," *J. Inf. Secur. Appl.*, vol. 58, no.5 , pp. 102779 , 2021.

[70] T. R. N. Rao and K.-H. Nam, "Private-key algebraic-code encryptions," *IEEE Trans. Inf. Theory*, vol. 35, no. 4, pp. 829–833, 1989.

[71] R. Rothblum, "Homomorphic encryption: From private-key to public-key," in *Theory of cryptography conference*, Springer, vol. 6597, no.3, pp. 219–234, 2011.

[72] D. E. Denning, "Digital signatures with RSA and other public-key cryptosystems," *Commun. ACM*, vol. 27, no. 4, pp. 388–392, 1984.

[73] A. A. Ayele and V. Sreenivasarao, "A modified RSA encryption technique based on multiple public keys," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 1, no. 4, pp. 859–864, 2013.

[74] R. Zuccherato, "Elliptic Curve Cryptography Support in Entrust," *Entrust ltd. Canada, Dated*, vol,2000, no. 5, pp. 1-8, 2000.

[75] D. Minoli and B. Occhiogrosso, "Blockchain mechanisms for IoT security," *Internet of Things*, vol. 1, no. 9, pp. 1–13, 2018.

[76] J. E. Silva, "An overview of cryptographic hash functions and their uses," *GIAC*, vol. 6 , no. 2, pp. 1, 2003.

[77] W.-Y. Tsai, T.-C. Chou, J.-L. Chen, Y.-W. Ma, and C.-J. Huang, "Blockchain as a platform for secure cloud computing services," in *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, IEEE, vol. 2020, no. 4, pp. 155–158, 2020.

[78] K. Gai, J. Guo, L. Zhu, and S. Yu, "Blockchain meets cloud computing: A survey," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 2009–2030, 2020.