Journal of Engineering, Science and Technology
ISSN: 2714-2396, Volume 7 Issue 1

SECURITY AND PRIVACY ISSUES OF DECENTRALIZED BLOCKCHAIN SYSTEMS

O. T. Banjo, O. Awodele, F.Y. Ayankoya

Department of Computer Science, Babcock University, Ilisan Remo, Ogun

Received Date: 1st May 2024 Accepted Date: 15th July 2024 Published Date: 15th Sept., 2024

ABSTRACT
Decentralized blockchain systems have gained popularity due to their potential for secure, transparent transactions without intermediaries, promoting trust and efficiency. However, they face security and privacy challenges, including cyber-attacks that can compromise data integrity and confidentiality. This literature review examines existing research on security and privacy issues in decentralized blockchain systems, highlighting critical challenges including 51% attacks, smart contract vulnerabilities, and user data exposure risks." Despite these challenges, cutting-edge solutions like advanced consensus algorithms and privacy-enhancing technologies offer promise for a more secure and reliable blockchain ecosystem. While decentralized blockchain systems hold promise for enhancing security and privacy, crucial hurdles must be overcome through continued investigation and innovation. Developing more resilient and scalable solutions is essential for widespread adoption. This review underscores the need for sustained research and innovation to address vulnerabilities and unlock the full potential of blockchain technology. Examining security and privacy challenges and solutions enables a more secure and reliable blockchain ecosystem to emerge.
Keywords: Decentralized blockchain systems, security, privacy, etc.

1. INTRODUCTION

Blockchain technology is transforming transactions by decentralizing trust, enabling secure, transparent, and efficient exchanges. Decentralized blockchain systems offer enhanced efficiency, reduced costs, and strengthened trust, but security and privacy concerns require ongoing investigation [1]. Security safeguards shield the system from attacks, while privacy measures protect user data from unauthorized access. Challenges include smart contract vulnerabilities, 51% attacks [2], and user data leaks, which have significant implications for adoption and sustainability. Research has identified attack vectors and proposed solutions like consensus algorithms (Proof of Stake (PoS) and Delegated Proof of Stake (DPoS)) [3] and privacy-enhancing technologies (Zero-Knowledge Proofs (ZKPs) and ring signatures). However, the effectiveness and scalability of these solutions require further evaluation, and continuous assessment is necessary to stay ahead of emerging threats.

2. OBJECTIVE

The objective is to identify and synthesize the findings from the existing literature on security and privacy challenges facing decentralized blockchain systems. This review identifies key challenges, highlights research gaps, and proposes future directions to address these gaps, enhancing understanding of security and privacy challenges in decentralized blockchain

systems and providing valuable insights into potential solutions for more secure and private blockchain technologies.

3. LITERATURE REVIEW ON DECENTRALIZED BLOCKCHAIN SYSTEM

3.1 *Blockchain system*

Blockchain is a pioneering, decentralized, and distributed digital ledger technology that enables secure, transparent, and tamper-proof recording of transactions across a network of computers, providing a robust, resilient, and permanent record of all transactions with unparalleled integrity:

"Blockchain technology operates on a consensus mechanism, which enables secure and transparent data sharing among participants, ensuring data validity and agreement without relying on a central authority. This decentralized approach facilitates trustless transactions, allowing participants to share data confidently and securely, without the need for intermediaries [4]. Blockchain technology, first introduced in the Bitcoin white paper, has evolved to impact various industries and technologies beyond cryptocurrency [5]. Blockchain Revolution" explores its transformative effects on finance, business and the world at large, offering insights into its far-reaching potential [1]. Its broader applications include secure data storage, supply chain management and decentralized transactions, making it a revolutionary force in the digital landscape The Ethereum Yellow Paper [6] describes how blockchain principles are applied to create a more general ledger of transactions, expanding potential use cases beyond financial transactions.

3.2 *Blockchain Categories*

There are mainly three types of blockchain systems based on their access rights and degree of decentralization:

i. Public blockchains are open, decentralized, and transparent, allowing anonymous participation. Examples include Bitcoin and Ethereum, which offer high security and decentralization, enabling trustless transactions and censorship-resistant networks. [6], [4].
ii. Private blockchains are secure, centralized networks with restricted access and participation, limited to trusted entities or individuals, ensuring high privacy, security, and efficiency in transactions and data sharing, ideal for enterprise and institutional use cases.
iii. Blockchain alliance: The Consortium blockchains are a hybrid model that combines the features of public and private blockchains. They are managed by a pre-selected group of nodes, often involving multiple organizations, allowing for a shared ledger with controlled access.
iv. hybrid blockchains: combine elements of public and private blockchains, offering flexibility and customization for specific use cases and requirements, enabling both decentralization and control.

3.3 *Decentralized blockchain system*

A decentralized blockchain system is a self-governing, distributed network, operating without central authority or intermediaries, where nodes collectively validate and verify transactions, ensuring a secure, transparent, and censorship-resistant ecosystem, free from single points of control. In this decentralized system, each node maintains a copy of the blockchain, ensuring transparency and immutability, with a shared ledger updated in real-

time. Decentralization provides a robust security framework, mitigating single-point failure risks and creating a resilient ecosystem virtually immune to data tampering and censorship, with identical copies across nodes ensuring consistency and trust. Examples of decentralized blockchain systems include Bitcoin and Ethereum, which utilize consensus mechanisms like Proof-of-Work (PoW) and Proof-of-Stake (PoS) to achieve node consensus and ensure network security and integrity.

4. SECURITY AND PRIVACY ISSUES /CHALLENGES FACING DECENTRALIZED BLOCK CHAIN SYSTEM

Decentralized blockchain systems face various security and privacy challenges, including scalability, consensus protocol vulnerabilities, smart contract vulnerabilities, privacy leakage, and attacks on network topology, among others [7]. In particular, scalability and privacy challenges have been identified as major impediments to the widespread adoption of blockchain technology. Decentralized blockchain systems face several security and privacy challenges that limit their widespread adoption. These challenges can be categorized into three main groups: scalability challenges, consensus protocol vulnerabilities, and smart contract vulnerabilities.

4.1 *Scalability Challenges*

Decentralized blockchain systems face a significant challenge in achieving scalability and increased transaction capacity. As the blockchain network grows, the number of transactions increases, leading to slower transaction speeds and higher fees, posing scalability challenges for the decentralized system. This limits the usability of blockchain technology in applications that require high throughput and low latency, such as financial transactions. Some of the most significant scalability challenges are:

1. Network congestion occurs when a blockchain's user base and transaction volume increase, causing slower processing times and higher fees, posing a significant scalability challenge.
2. Block size limitations in blockchain systems restrict the number of transactions per block, leading to delayed confirmation times, particularly during high network activity, hindering scalability and overall efficiency.
3. Many blockchain systems use resource-intensive consensus mechanisms, such as proof-of-work, limits transaction processing capacity, creating scalability challenges and restricting network efficiency and speed.
4. Blockchain systems require vast storage for a comprehensive transaction record, leading to increasing storage demands as transactions multiply, making it challenging for smaller nodes to participate and maintain a full copy of the blockchain.
5. Interoperability: Many blockchain systems are not interoperable, which means that transactions between different blockchains require intermediaries and additional processing time, which can slow down the entire network.

4.2 *Consensus Protocol Vulnerabilities*

Another major challenge facing decentralized blockchain systems is the vulnerability of their consensus protocols to attacks. Consensus protocols ensure agreement on the blockchain's state, and if compromised, the blockchain's security and integrity are at risk, potentially leading to fraudulent activities and network instability [8].

One of the most significant threats to consensus protocols is the 51% attack, where a malicious actor gains control of more than half of the network's computational power,

enabling them to manipulate the blockchain's state and compromise its integrity [9]. Other attacks include the selfish mining attack, where a miner can increase their chances of mining a block by withholding their newly mined blocks, and the double-spending attack, where an attacker can spend the same cryptocurrency twice. Some of the most common consensus protocol vulnerabilities that decentralized blockchain systems face include:

1. 51% Attack: In a proof-of-work consensus protocol, a 51% attack occurs when a single entity or group controls more than 51% of the network's computational power. This allows the attacker to take control of the network and manipulate the ledger.
2. A Sybil attack involves a single entity creating multiple fake identities to manipulate the consensus protocol and gain control over the blockchain network's decision-making process. This can be used to manipulate the ledger, double-spend, or prevent legitimate transactions from being processed.
3. Forking: Forking occurs when a blockchain splits into two separate chains, creating two different versions of the ledger. This can occur due to differences in consensus rules or a disagreement between participants in the network.
4. A long-range attack involves an attacker using old valid blocks to create a fork, rewriting the blockchain's history and potentially altering its integrity. This can be done by accumulating computational power over time or by obtaining access to older private keys.
5. Time jacking: Time jacking involves manipulating a block's timestamp to deceive the network into accepting an invalid block, compromising the blockchain's integrity and security.

4.3 *Smart Contract Vulnerabilities*

Another significant issue confronting decentralized blockchain systems is the ability to attack their consensus procedures. Consensus procedures ensure agreement on the blockchain's state, and if compromised, the blockchain's security, integrity, and overall trustworthiness can be severely jeopardized and potentially exploited [8]. The DAO hack, which exploited a smart contract flaw to steal millions of dollars' worth of bitcoin, is a notable example of smart contract vulnerabilities and their potentially devastating consequences [10]. The following are some of the most typical smart contract flaws that decentralized blockchain systems encounter:

1. Reentrancy: Reentrancy occurs when a contract calls itself before the first invocation has been completed, allowing an attacker to execute malicious code repeatedly. Reentrancy attacks, where an attacker can repeatedly call a vulnerable function in a smart contract, and timestamp dependence attacks [11], where an attacker can manipulate the timing of smart contract execution to their advantage.
2. Integer overflow and underflow occur when a value surpasses the maximum or minimum limit of its data type, causing errors and potential security vulnerabilities. This can cause unexpected behavior in a smart contract, allowing an attacker to manipulate the contract's state.
3. Denial-of-Service (DoS) attacks occur when an attacker floods a contract with numerous transactions or requests, overwhelming it and preventing legitimate users from accessing the contract.
4. Malicious Libraries: Smart contracts can include libraries that provide additional functionality. However, if these libraries are malicious, they can exploit the contract's vulnerabilities and manipulate the contract's state.

5. Timestamp Dependency: Some smart contracts rely on timestamps to perform specific functions, such as releasing funds. However, timestamps can be manipulated by an attacker, allowing them to manipulate the contract's behavior.

Despite these proposed solutions, scalability remains a significant challenge for decentralized blockchain systems. The limitations of current solutions, such as the high computational and storage requirements of sharding techniques, highlight the need for further research into more efficient and scalable approaches [12].

5. SOLUTIONS TO SECURITY AND PRIVACY ISSUES /CHALLENGES FACING DECENTRALIZED BLOCKCHAIN SYSTEM

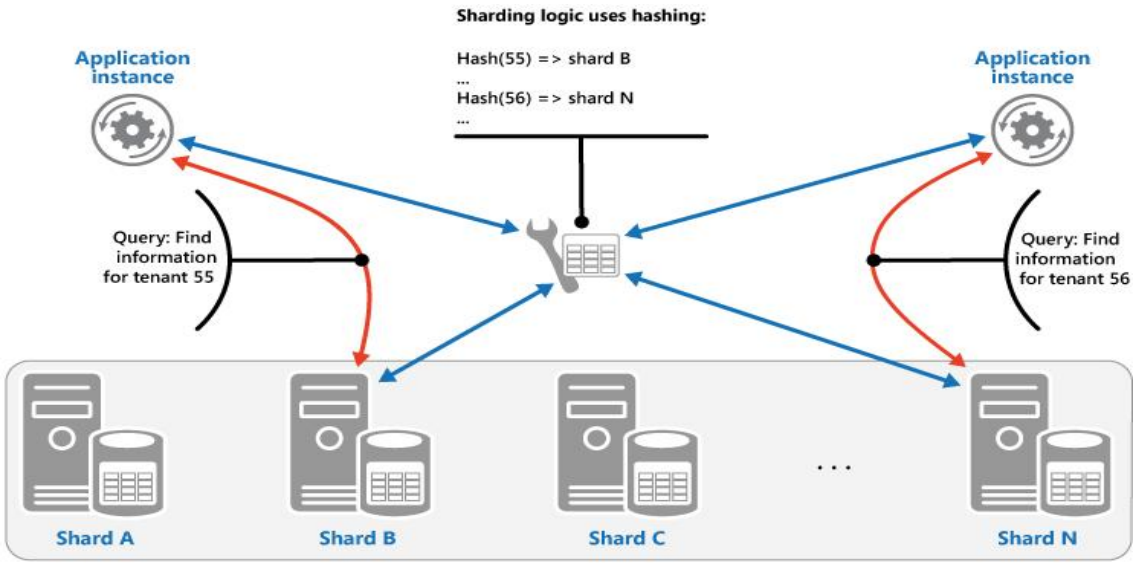
Decentralized blockchain systems hold enormous potential for disrupting various industries, but security and privacy challenges must be effectively addressed to unlock their full capabilities. The blockchain community can establish secure and private decentralized ecosystems by leveraging innovative solutions like alternative consensus mechanisms, formal verification, advanced identity management, and privacy-enhancing techniques like zero-knowledge proofs for a safer future

5.1 Solution to scalability challenges

Sharding Techniques divides the blockchain into smaller, independent sub-chains, increases overall network capacity, providing a potential solution to scalability issues and enhancing blockchain efficiency. Sharding divides the blockchain into smaller sub-chains, increasing manageability and reducing congestion in the network. Off-chain scaling solutions like the Lightning Network process transactions faster and cheaper by bypassing the blockchain, significantly enhancing overall network efficiency and scalability.

Sharding is dividing the blockchain network into smaller, more manageable sub-chains that can process transactions concurrently. Sharding can potentially increase blockchain networks' transaction throughput substantially but requires additional computational and storage resources to maintain consistency, making it computationally expensive. Sharding enhances the system's overall speed and efficiency by enabling simultaneous processing of multiple transactions, improving the blockchain network's performance and scalability. Sharding also reduces each node's storage requirements, making it easier for new nodes to join the network.

However, sharding also introduces new challenges to the blockchain system. One challenge is maintaining the security and consistency of the network, as each node is only responsible for verifying transactions on its assigned shard. his necessitates meticulous coordination and communication among nodes to guarantee the network's security and consistency, ensuring a unified and reliable blockchain ecosystem. Another challenge is ensuring that transactions are properly routed to the correct shard. This requires a mechanism for determining which shard a particular transaction belongs to, and ensuring that the correct node processes it. Different rules are applied to the shard key by various database sharding techniques to identify the proper node for a given data row. These are some typical sharding architectures.



Sharding technique for breaking down data into smaller and more manageable sub chains [13].

1. Range Sharding

Database rows are divided using range-based sharding, also known as dynamic sharding. The database designer then gives the appropriate range a shard key. For example, the database designer divides the data into the following categories based on the first letter of the student's name.

Student Name first Letter	Shard key
Start with A to I	A to I
Start with J to S	J to S
Starts with T to Z	T to Z

The application checks the student's name to determine the correct shard key then matches the key to its physical node and stores the row on that machine. The application also performs a reverse match when searching for a particular record.

Cons and benefits

Range-based sharding may lead to data saturation on a single physical node, depending on the data contents. In our illustration, shard A (which contains names that begin with A to I) may have a lot more data rows than shard C. (containing names that start with T to Z). It is simpler to implement, though.

2. Harsh sharding

A hash function is a mathematical formula used in hashed sharding to assign a shard key to each database entry, enabling efficient data allocation. The hash function creates a hash value using the data from the row. The program stores the data in the corresponding physical shard using the hash value as a shard key.

Hashed sharding is a technique used by software developers to uniformly distribute data among several shards in a database. As an illustration, the program divides student records into two shards with different hash values of 1 and 2.

Name	Hash Value
Matthew	1
Bernice	2
Martins	1
Grace	2

Cons and benefits
Hashed sharding distributes data evenly among the physical shards, but it does not split the database according to the information's significance. Therefore, if more physical shards are added to the computing environment, software developers may have problem reassigning the hash value.

5.2 *Consensus Protocol Solutions and Countermeasures*

Consensus protocols are mechanisms used in distributed systems to achieve agreement among nodes on a shared network state, ensuring consistency and unity. Consensus protocols play a crucial role in ensuring the integrity and security of blockchain systems, preventing attacks like double-spending and maintaining the trustworthiness of the network.
Several consensus protocol solutions exist, each with unique strengths and weaknesses, including popular options like Proof of Work and Proof of Stake:

1. Proof of Work (PoW) involves nodes competing to solve a cryptographic puzzle, with the first solver rewarded with block addition rights. PoW is secure but has high energy consumption and 51% attack vulnerability.
2. Proof of Stake (PoS) selects nodes to add blocks based on their cryptocurrency holdings, reducing energy consumption but potentially leading to centralization and decreased network decentralization.
3. Delegated Proof of Stake (DPoS) involves nodes voting for delegates to add blocks, increasing efficiency, but also risking centralization, as a small group of elected delegates controls the blockchain's development and validation processes.
4. Practical Byzantine Fault Tolerance (PBFT) is a consensus algorithm that enables nodes to reach consensus despite faulty or malicious nodes, ensuring reliability and security in permissioned blockchain systems and distributed networks.
5. Raft: Raft is a consensus algorithm that elects a leader node that is responsible for adding new entries to the log. It is designed for high availability and is commonly used in distributed systems.

Countermeasures against attacks on consensus protocols include:

1. Adding more nodes to the network increases decentralization, making it harder for attackers to gain control, and enhancing network security and resilience.
2. Reducing the rewards for attacking the network: This can be done by reducing the block rewards or by implementing penalties for malicious behavior.
3. Multi-signature transactions require multiple signatures, preventing attacks like double-spending, and enhancing security by ensuring multiple parties validate and authorize each transaction.
4. Increasing the difficulty of the cryptographic puzzles: This can make it more difficult for attackers to solve the puzzle and gain control of the network.
5. Checkpointing involves periodically storing the network's state, creating a permanent record, making it harder for attackers to alter the blockchain's history, and enhancing network security and integrity.

5.3 *Smart Contract Solutions and Countermeasures*

Smart contracts are self-executing digital agreements that automate the process of verifying, executing, and enforcing contract terms, ensuring efficient and secure transactions. Smart contracts operate on a blockchain network, ensuring security, transparency, and tamper-proof

execution, with diverse applications across finance, real estate, healthcare, and supply chain management, transforming various industries with increased efficiency and trust.
To address smart contract vulnerabilities, solutions include formal verification, auditing, testing, and bug bounties, ensuring secure execution

Formal Verification Techniques
Formal verification uses mathematical techniques to rigorously prove software correctness, ensuring error-free execution and enhancing overall system reliability. Formal verification techniques can be used to ensure the correctness of smart contract code, identifying potential vulnerabilities and ensuring secure execution. However, formal verification techniques can be computationally expensive and require significant expertise. Some of the common formal verification techniques used in smart contract solutions and countermeasures are:

1. Model checking: Model checking is a technique that automatically checks if a smart contract model meets its requirements. It involves creating a finite state machine model and checking it against a set of properties to ensure it's free from vulnerabilities.
2. Theorem proving involves using mathematical proofs to verify the correctness of smart contracts, utilizing a theorem prover to ensure the contract meets its specified requirements, guaranteeing accuracy and security.
3. Static analysis: Static analysis is a technique for analyzing smart contract code without execution, identifying vulnerabilities and ensuring compliance with specified requirements, using rules or specifications.
4. Type checking: Type checking checks variables and functions in smart contracts to identify vulnerabilities and ensure compliance with requirements, preventing potential breaches.
5. Automated testing: Automated testing verifies smart contract functionality, correctness, and potential vulnerabilities, ensuring compliance with specified requirements and identifying potential vulnerabilities.

In summary, Formal verification techniques ensure smart contract security and correctness by creating a formal model and logical reasoning to identify vulnerabilities, ensuring reliability and trustworthiness.

6. CONCLUSION

This review has provided a comprehensive overview of the security and privacy challenges facing decentralized blockchain systems. It has identified key challenges, evaluated existing solutions, and analyzed their effectiveness. The review has demonstrated the need for a robust and secure blockchain framework to ensure the integrity and confidentiality of transactions.

7. RECOMMENDATION

To address the challenges identified in this review, it is recommended that future decentralized blockchain systems prioritize privacy, adopt advanced consensus mechanisms like Proof of Stake, and implement zero-knowledge proofs by default. Additionally, scalability can be enhanced through Layer 2 solutions like the Lightning Network, and interoperability can be promoted through standards for seamless data and asset exchange. Long-term security can be ensured with quantum-safe cryptography, and user privacy can be protected with a decentralized identity management system. Collaboration with regulators,

regular security reviews, continuous innovation, and user education are also essential for ensuring the security and privacy of decentralized blockchain systems.

REFERENCES

[1] Joshi, S., Pise, A. A., Shrivastava, M., Revathy, C., Kumar, H., Alsetoohy, O., &Akwafo, R. "Adoption of Blockchain Technology for Privacy and Security in the Context of Industry 4.0." *Wireless Communications and Mobile Computing*, *2022*, e4079781.(2022). https://doi.org/10.1155/2022/4079781

[2] Bhushan, B., Sinha, P., Sagayam, K. M., & J, A. "Untangling blockchain technology: A survey on state of the art, security threats, privacy services, applications and future research directions." *Computers & Electrical Engineering*, 106897.(2020). https://doi.org/10.1016/j.compeleceng.2020.106897

[3] Yang, F., Zhou, W., Wu, Q., Long, R., Xiong, N. N., & Zhou, M. "Delegated Proof of Stake with Downgrade: A Secure and Efficient Blockchain Consensus Algorithm With Downgrade Mechanism." *IEEE Access*, *7*, 118541–118555. (2019). https://doi.org/10.1109/access.2019.2935149

[4] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. (2008). Retrieved from: https://bitcoin.org/bitcoin.pdf

[5] Tapscott, D., & Tapscott, A. Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World. Portfolio. (2016).

[6] Wood, D. D. "*ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER*." Undefined. (2014). https://www.semanticscholar.org/paper/Ethereum%3a-A-Secure-Decentralised-Generalised-Ledger Wood/da082d8dcb56ade3c632428bfccb88ded0493214

[7] Zhang, P., & Zhou, M. "Security and Trust in Blockchains: Architecture, Key Technologies, and Open Issues." *IEEE Transactions on Computational Social Systems*, *7*(3), 790–801. (2020). https://doi.org/10.1109/TCSS.2020.2990103.

[8] Akbar, N. A., Muneer, A., ElHakim, N., &Fati, S. M. "Distributed Hybrid Double-Spending Attack Prevention Mechanism for Proof-of-Work and Proof-of-Stake Blockchain Consensuses." *Future Internet*, *13*(11), 285. (2021). https://doi.org/10.3390/fi13110285

[9] Guru, A., Mohanta, B. K., Mohapatra, H., Al-Turjman, F., Altrjman, C., & Yadav, A. "A Survey on Consensus Protocols and Attacks on Blockchain Technology." *Applied Sciences*, *13*(4), 2604. (2023). https://doi.org/10.3390/app13042604

[10] Samreen, N. F., &Alalfi, M. H. "A Survey of Security Vulnerabilities in Ethereum Smart Contracts." *ArXiv:2105.06974 [Cs]*.(2021). https://arxiv.org/abs/2105.06974

[11] Praitheeshan, P., Pan, L., Yu, J., Liu, J., & Doss, R. "Security Analysis Methods on Ethereum Smart Contract Vulnerabilities: A Survey." *ArXiv:1908.08605 [Cs]*. (2020). https://arxiv.org/abs/1908.08605

[12] Joshi, V. "*Scalability Problems: Hidden Challenges of Growing a System*." Medium. (2019, February 27). https://medium.com/baseds/scalability-problems-hidden-challenges-of-growing-a-system-f74313b063c3

[13] Martinekuan. "*Sharding pattern - Azure Architecture Center*." Learn.microsoft.com. (2018). https://learn.microsoft.com/en-us/azure/architecture/patterns/sharding