Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 07-2024
DOI: 10.5281/zenodo.12743067

BLOCK EASE: INTRODUCING THE LAZY BLOCKCHAIN APPROACH FOR EFFICIENT DATA INTEGRITY VERIFICATION IN CLOUD-BASED E-GOVERNANCE

RATNESH KUMAR DIXIT
Research Scholar, Invertis University, Bareilly, Uttar Pradesh, India. E-mail: ratneshbalramdixit@gmail.com

R K SHUKLA
Faculty of Engineering and Technology, Invertis University, Bareilly, Uttar Pradesh, India.
E-mail: rkshukla30@gmail.com

NAYANEESH KUMAR MISHRA
Associate Professor, Department of Computer Science, LDC Institute of Technical Studies, Prayagraj, Uttar Pradesh, India. E-mail: nayaneesh@gmail.com

RATNESH MISHRA
Assistant Professor, Computer Science & Engineering, BIT, Mesra Patna Campus, Bihar, India.
E-mail: r.mishra@bitmesra.ac.in

RAVI SHANKAR SHUKLA
Assistant Professor, Department of Computer Science, Saudi Electronics University, KSA.
E-mail: ravipraful@gmail.com

Abstract

In data-intensive cloud applications, maintaining data integrity amid frequent data block modifications poses a significant challenge. Despite employing blockchain to track data file updates, delays occur as blocks are added only upon majority consensus, causing update delays. To address this, we propose a Lazy Blockchain methodology that efficiently records updates. By tracking transactions and system state commitments, blocks are added lazily post verification, accommodating the rapid update frequency. Our methodology ensures integrity checks and has demonstrated efficiency across various parameters. The results highlight the solution's efficacy in managing data updates while maintaining integrity.

Keywords: Cloud Computing, Data Security, Blockchain, Data Integrity, E-Governance, Lazy, Blockchain.

INTRODUCTION

The increasing prevalence of data-intensive applications has raised significant concerns about data integrity and security. Given the substantial volumes of data generated and processed by these applications, ensuring the integrity of such data is of paramount importance. Blockchain technology emerges as a promising solution to address these concerns, providing a distributed and immutable ledger where transactions are cryptographically linked in blocks, establishing a robust mechanism for securing data [1, 1, 4–6, 8, 9].

Blockchain is being used for ensuring data integrity in files. When the updates are made to the file, the file update transactions are added as block to the blockchain

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 07-2024
DOI: 10.5281/zenodo.12743067

ensuring data integrity. Any user can check the integrity of the file updates using the blockchain. Using blockchain to preserve the transaction updates to a file in cloud is easy and feasible when the frequency of the file updates is nearly equal to the time taken to add a new block the blockchain.

This is particularly relevant because the time taken to add a new block to the blockchain is considerable. If the frequency of the updates is more, then the the time taken to add a new block will become the bottleneck for the newer updates to take place as they cannot be implemented until and unless the previous blocks get added to the blockchain. Therefore, as the scale and complexity of data-intensive applications continue to grow, finding more efficient ways to update data within a blockchain becomes crucial.

To address this challenge, our research introduces a novel approach using a technique called Lazy Hashing Technique. This technique aims to streamline the process of updating data blocks within a blockchain while still maintaining the fundamental principles of data integrity. The core idea behind Lazy Hashing is to create a lazy copy of the original data block and keep adding it to the blockchain without the integrity check and without the application of consensus algorithm. This tree will call as Dirty Tree.

However, in the background, the blocks in the dirty tree are allowed to go through consensus algorithm with proper integrity check and are added to the blockchain only after they pass the requisite checks. By doing so, we can reduce the computational burden associated with addition of the blockchain and still making data block updates more efficient. This calls this approach as Block Ease.

In summary, our research presents the Block Ease as an innovative method to address the computational challenges associated with data block updates in blockchain technology. By adopting this approach, we aim to enhance the efficiency of data- intensive applications while ensuring data integrity which remains a top priority. This technique represents a significant step towards optimizing the performance of blockchain-based systems in the context of evolving data-intensive applications on cloud.

BACKGROUND

The concept of blockchain technology has revolutionized the way data is recorded and verified, gaining widespread acclaim for its secure and tamper-resistant nature. Built on principles of decentralization and immutability, blockchain operates as a distributed ledger, forming a chain of interconnected data blocks.

Critical to maintaining data integrity within each block is the use of cryptographic hashes. Hashing, a process that converts variable-size input into a fixed-size output, ensures the uniqueness of data representations. Even minor alterations in input result in distinct hash values, thus safeguarding data integrity within the blockchain.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 07-2024
DOI: 10.5281/zenodo.12743067

An inherent advantage of blockchain lies in its immutability. Once data is recorded and added to a block, attempts to modify or erase it without detection are highly improbable. This resilience stems from cryptographic linkages between blocks, where each block's hash is computed based on its data and the previous block's hash. Any alteration to block data disrupts the entire chain, alerting all network participants to potential tampering.

Further enhancing reliability and integrity is blockchain's decentralized nature. Data is not confined to a single central repository but is distributed across a network of nodes, ensuring collective validation and recording of transactions. This decentralized architecture mitigates the risk of manipulation or single-point failures, fostering transparency and trust among network participants.

In essence, blockchain technology relies on a distributed ledger and cryptographic hashing to ensure data integrity and security. Its transparent and decentralized design underscores the reliability and trustworthiness of blockchain networks.

RELATED WORKS

Naresh Vurukonda and B. Thirumala Rao conducted a comprehensive study [11] aimed at identifying security concerns prevalent in cloud computing. Their investigation focused on issues pertaining to data storage, access control, and identity management, while also proposing potential solutions to mitigate these challenges [10]. Contributing to the field, Ayesha Malik and Muhammad Mohsin Nazir devel- oped a robust security architecture tailored for cloud service providers. Their work encompassed detailed descriptions of various cloud service models, emphasizing the protection of users' confidential and sensitive information [7].

Delving into data protection and storage security within cloud computing, Yunchuan Sun, Junsheng Zhang, Yongping Xiong, and Guangyu Zhu conducted a thorough analysis of existing security mechanisms [12]. Their research shed light on the complexities of information security in the cloud environment. In a comprehensive review, Mazhar Ali, Samee U. Khan, and Athanasios V. Vasi- lakos discussed the prevalent security issues in cloud computing and highlighted recent solutions and mechanisms addressing these concerns [3].

Sultan Aldossary and William Allen explored the nuances of cloud data storage challenges and potential remedies. Their research encompassed discussions on data integrity, confidentiality, availability, virtualization, and the enumeration of various threats within cloud computing [2].

PROPOSED LAZY HASHING TECHNIQUE

In this section, we elaborate upon our proposed block Ease Technique 1 for the blockchains for maintaining data integrity at cloud servers with highly frequent updates. This method is effective in cases where the data blocks are secured using

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 07-2024
DOI: 10.5281/zenodo.12743067

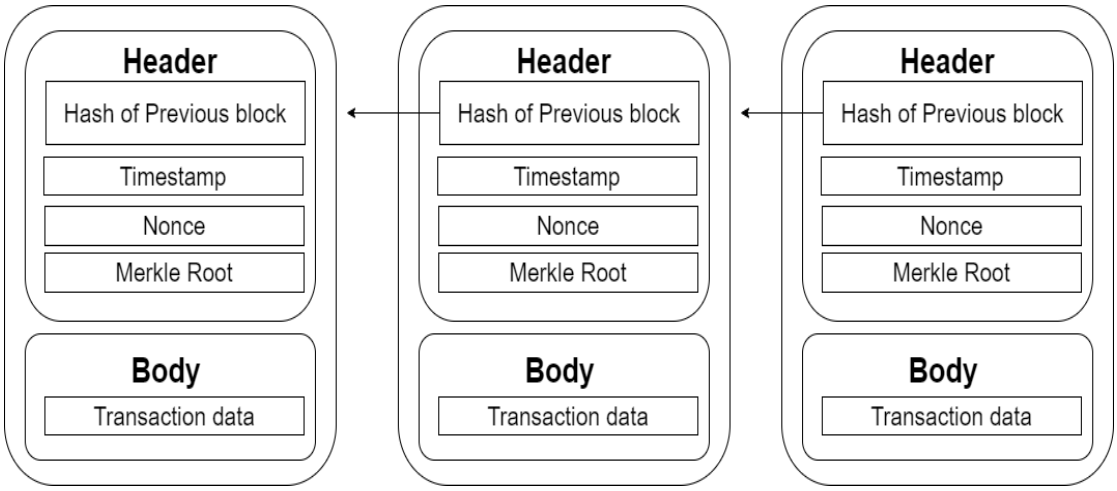

Fig 1: block Ease: The figure shows the structure of the blockchain obtained using the block Ease methodology

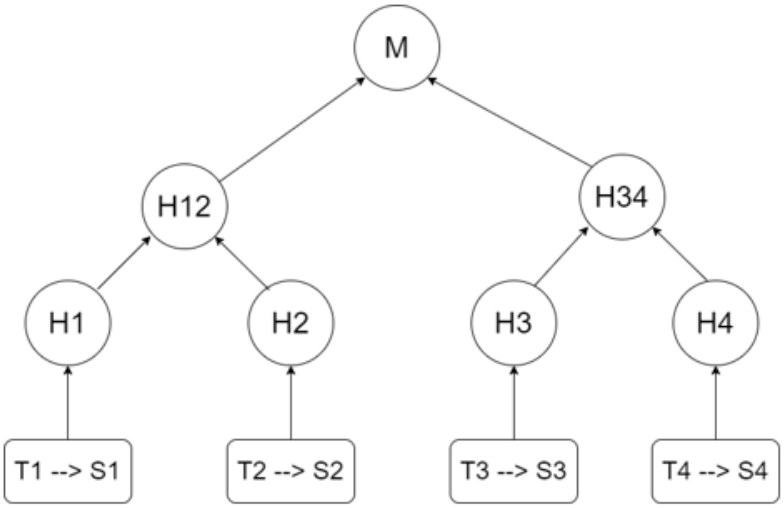

Fig 2: Merkle tree showing the transactions and the state commitments. If T1 is the transaction then S1 is the state commitment of the file after the transaction T1. M is the Merkle root.

Blockchain and the updates to the data blocks are frequent. In such cases, addition of new blocks in the blockchain becomes computationally expensive and also behaves like a bottleneck for the upcoming updates. Our proposed block Ease prepares a dirty tree containing all the file update transactions in the same sequence as they actually occur. The dirty tree is called so because the blocks are added without any integrity check and without the application of consensus algorithm. Such blocks that get added without any integrity check are called Dirty nodes and the tree is called Dirty Tree. Each transaction also contains a hash code of the state of the file after that file update transaction.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 07-2024
DOI: 10.5281/zenodo.12743067

During the addition of the dirty node as a block in the blockchain, the consensus mechanism is applied. The state commitment in the form of the hash code of the file after the transaction is applied, is checked for integrity check by the various nodes before the consensus mechanism allows or rejects the addition of the block to the blockchain. In case, the block rejects the dirty node for addition into the blockchain, the illegal transactions are ignored and the process is repeated for the next transactions and blocks in line.

To explain the steps of the proposed solution, we take a simple assumption that a file is transformed by 6 transactions namely T1, T2, T3, T4, T5 and T6. The state commitments corresponding to the transactions T1, T2, T3, T4, T5 and T6 are S1, S2, S3, S4, S5 and S6 respectively. A block can be formed using the threshold of 3 transactions. This means that after T1, T2 and T3 transactions are complete and their integrity check verified, a block containing T1, T2, and T3 can be added in the blockchain.

Below are the steps for the Block Ease technique:

Accepting Update Requests

Block Ease starts by implementing the transaction issued by the user on the file. After the transaction is implemented, the transaction and the corresponding state commitment produced by hashing the file after the transactional update of the file are recorded in the same sequence as the original order of the transactions. This can be written as:

$$T1 -> S1,\ T2 -> S2,\ T3 -> S3$$

Similarly other transactions are also noted and recorded along with the state commitments in a particular sequence.

Creating a Dirty Data Block

When the number of recorded transactions reach a threshold number of values, they are collected to form a dirty block. For the demonstration purpose, we consider a threshold of three. Hence, the transactions T1, T2 and T3 will be collected to form a dirty block. The block contains the same structure as a standard blockchain.

A dirty block will contain the following components:

1. **Block Header**: This includes metadata about the block, such as the block number, timestamp (when the block was created), and a reference to the previous block's hash.
2. **Transactions**: These are the actual data entries or transactions that are being recorded on the blockchain. Each block can contain multiple transactions, depending on the blockchain protocol and network capacity. In the present example transactions T1, T2 and T3 are recorded and based on these 3 transactions a Merkle tree is constructed and added to the block. The structure of the Merkle tree is as shown in 2.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 07-2024
DOI: 10.5281/zenodo.12743067

3. **Merkle Root**: A Merkle root is a cryptographic hash of all the transactions included in the block. It serves as a summary or fingerprint of all the transactions in the block.
4. **Nonce**: A nonce is a random number used in the process of mining (for proof- of-work consensus algorithms) to generate a hash value that meets certain criteria, such as having a specific number of leading zeros.
5. **Block Hash**: The block hash is a cryptographic hash of the entire block, including the block header, transactions, Merkle root, and nonce. It uniquely identifies the block on the blockchain and is used to link it to the previous block.
6. However, it must be remembered that the consensus algorithm is not implemented as in case of standard blockchain. The block generated here is a dirty block.

Implementation of Consensus Mechanism to verify the blocks

While the transactions are being implemented and dirty blocks are being added, the consensus mechanism runs in parallel mode independent of the addition of the dirty blocks. This process runs slowly at its own pace and hence this does not have any impact on the frequency of the file updates. The implementation of the consensus mechanism verifies the integrity of each of the transactions. The previous state commitments and the transaction is used to verify the present state commitment. For example, in the case:

$$T1 \rightarrow S1, T2 \rightarrow S2, T3 \rightarrow S3$$

Transaction T2 is verified by implementing the transaction T2 on the state commitment S1 and then verifying whether it reaches the state commitment S2. When this is verified by the majority of the nodes using the consensus mechanism, the dirty block is added as a block to the final blockchain. Those transactions which do not qualify the consensus mechanism are ignored and left out from the final block in the blockchain.

Implementation and Methodology

To access the effectiveness of the proposed Block Ease methodology for preserving the data integrity in cloud environment, we designed an experiment. The experiment was designed using the AWS cloud service. The Google cloud service was used for hosting the experimental setup. Hyperledger Fabric was chosen as the blockchain framework. The data file was kept on the Google cloud storage facility. A smart contract was also installed to manage file updates and hash calculations on the blockchain network. Multiple users were simulated so that they can make multiple updates simultaneously. The hash for each updated file was calculated using a cryptography hashing algorithm SHA-256. We have set a threshold of 1000 updates before a block is added to the blockchain. The Block Ease adds the block using the proposed lazy blockchain method as described in the proposed methodology section. However, while the block is being added to the blockchain, newer file updates are updated to the file as they come and the hash value of the file is saved in the same

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 07-2024
DOI: 10.5281/zenodo.12743067

order. The addition of the block to the blockchain happens periodically rather than immediately after each update.

In the experiment, the integrity of file updates is validated by comparing the calculated hashes with the corresponding hashes stored in the blockchain. The consistency of blockchain records is verified to ensure that no unauthorized modifications have occurred.

The performance of the blockchain network is monitored by observing parameters like transaction throughput, latency, and resource utilization. The performance metrics

Table 1: This is Data Integrity Table. The table shows the list of integrity verification done using the blockchain.

File Update Number	Hash Calculated	Hash Stored on Blockchain	Integrity Verification
1	abc123...	abc123...	Passed
2	def456...	def456...	Passed
3	ghi789...	ghi789...	Passed
4	jkl012...	jkl012...	Passed
5	mno345...	mno345...	Passed

Table 2: Blockchain Consensus Mechanism: The table shows the number of transactions processed and the latency in processing the transactions.

Transaction Batch	Transactions Processed	Latency (Seconds)	Block Creation Time
1	100	0.2	10:00:05 AM
2	150	0.3	10:05:10 AM
3	120	0.25	10:10:15 AM
4	200	0.4	10:15:20 AM
5	130	0.35	10:20:25 AM

Table 3: This table shows the overhead percentage are analysed to identify any bottlenecks or scalability issues in the lazy blockchain process.

Data Size (MB)	Overhead (%)
10	1.2
20	1.8
30	2.0
40	2.3
50	2.5

Experimental Results and Discussion

The results of the experiment for lazy blockchain are compiled and shown in tables. The results are compiled for different parameters such as for Integrity verification, Latency in addition of blocks to the blockchain, transaction throughput, CPU and memory usage in percentage, and fault tolerance.

Once we look at the results, it is very clear that the proposed lazy blockchain method is working fine. The block creation time for each batch of transactions is consistent,

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 07-2024
DOI: 10.5281/zenodo.12743067

indicating that the lazy blockchain mechanism effectively manages block creation intervals. The consistency of block creation and the maintenance of transaction order within blocks suggest that the lazy blockchain mechanism maintains data integrity and security effectively.

The integrity of the user data can be verified easily and the integrity check is giving correct results regarding the data updates as seen in the table 6. The data hash held with the user is being matched with the hash in the blockchain to verify the integrity of the data update. The observed latency for block creation ranges from 0.2 to 0.4 seconds as can be seen in Table 2, which suggests relatively low latency in adding new blocks. This indicates efficient block propagation.

Table 4: Scalability and Performance: The table shows the transaction throughput in comparison to the number of transactions being input. The table also shows the CPU and the memory usage in percentage.

Number of Transactions	Transaction Throughput (Transactions per Second)	CPU Usage (%)	Memory Usage (%)
1000	800	60	50
1500	1200	70	60
2000	1500	75	65
2500	1800	80	70
3000	2000	85	75

Time (Transactions)	Data Consistency (%)
0	100
100	99.9
200	99.9
300	99.8
400	99.8

Rand validation processes within the network. The average low latency also tells about the efficiency of the Consensus Mechanism.

By analyzing the number of transactions processed per batch, the transaction throughput of the lazy blockchain mechanism can be evaluated. As seen in Table 4, the lazy blockchain shows higher transaction volumes processed per batch. May indicate improved throughput and network efficiency. The highest transaction throughput is 80% when the 800 out of 1000 input transactions are processed by the blockchain clearly indicating that the blockchain is working efficiently while maintaining the data integrity. It is also observed that the lazy mechanism is efficiently able to aggregate transactions into blocks based on predefined thresholds by taking sufficient time. The delay in adding the blocks avoids the overhead of immediate block creation for each transaction. This approach has helped in reducing network congestion and optimizing resource utilization. The resource utilization Table 4 clearly shows the optimum usage of resources with signs of over utilization of CPU and memory. From the scalability point of view, it can be easily said that mechanism can easily handle different volumes of transaction without much change in the latency. This tells us that the proposed system is easily scalable.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 07-2024
DOI: 10.5281/zenodo.12743067

Advantages of the Block Ease

The block Ease method offers several advantages for ensuring the integrity of files, especially in scenarios with high update frequencies:

1. **Efficiency**: block Ease allows for efficient handling of frequent file updates by recording state commitments after each transaction. This ensures that the update process is not hindered by delays typically associated with consensus mechanisms.
2. **Real-time Updates:** With block Ease, files can be updated in real-time without waiting for the completion of the consensus mechanism. This enables users to maintain up-to-date data without sacrificing efficiency.
3. **Reduced Latency:** By introducing dirty blocks that bypass the consensus mechanism initially, block Ease significantly reduces latency in the update process. This ensures that file updates occur promptly, enhancing overall system responsive- ness.
4. **Scalability**: The parallel verification of dirty blocks through the consensus mechanism allows for scalability in handling high update frequencies. Block Ease can accommodate increased transaction volumes without sacrificing performance.
5. **Data Integrity**: Despite the expedited update process, block Ease maintains data integrity by subjecting dirty blocks to the consensus mechanism for verification. This ensures that only valid transactions are added to the final blockchain, preserving the integrity of the entire dataset.
6. **Decentralization**: block Ease retains the decentralized nature of blockchain technology by employing consensus mechanisms for verification. This decentralization ensures trust and transparency in the update process, as multiple nodes participate in the validation of transactions.
7. **Resilience to Network Congestion**: In situations where network congestion may occur due to high update frequencies, block Ease remains resilient. The method allows for the smooth processing of transactions without being bottlenecked by network limitations.
8. **Flexibility**: block Ease offers flexibility in adapting to diverse use cases and applications with varying update requirements. Its design allows for customization to suit specific needs while maintaining the core principles of data integrity and efficiency.
9. **Enhanced User Experience**: By ensuring swift and reliable file updates, block Ease enhances the overall user experience. Users can interact with the system seamlessly, confident in the integrity and timeliness of their data.
10. **Innovation**: Block Ease introduces a novel approach to file integrity maintenance in blockchain systems. Its innovative methodology opens doors for further advancements in blockchain technology, paving the way for more efficient and scalable solutions in data management and integrity verification.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 07-2024
DOI: 10.5281/zenodo.12743067

Potential Use Cases

BlockEase, with its innovative approach to ensuring the integrity of files in environment characterized by high update frequencies, holds promise across various industries and applications. In the financial sector, BlockEase can revolutionize transaction processing and auditing, facilitating real-time verification of financial data while adhering to regulatory standards. Moreover, in healthcare, BlockEase can bolster the security and reliability of electronic health records (EHRs), ensuring that patient data remains accurate and confidential in dynamic healthcare environments. In supply chain management, BlockEase offers the potential to streamline processes by providing transparent and traceable updates on inventory, shipments, and transactions, thereby mitigating fraud and enhancing product authenticity. Furthermore, government agencies can leverage BlockEase to maintain the integrity of public records, such as land titles and voting records, ensuring transparency and accountability. In the realm of intellectual property protection, BlockEase can safeguard digital assets by validating copyrights, patents, and digital media files. Its application extends to supporting smart contracts and decentralized applications (DApps), enabling secure and efficient execution of contractual agreements and transactions. Additionally, BlockEase container streamline insurance claims processing, verify academic records, and enhance credential authentication across various domains. These diverse applications underscore the versatility and potential impact of BlockEase in ensuring data integrity and security in an increasingly digital and interconnected world.

CONCLUSION

In conclusion, BlockEase emerges as a transformative solution for addressing the challenges associated with ensuring file integrity in environments characterized by high update frequencies. By introducing a novel methodology that balances efficiency and security, BlockEase offers a versatile platform with applications across diverse industries and domains. Its ability to streamline processes, enhance transparency, and mitigate risks associated with data tampering makes it a valuable asset in sectors such as finance, healthcare, supply chain management, and government services. With BlockEase, organizations can embrace real-time updates without compromising on data integrity, thereby unlocking new opportunities for innovation and efficiency. As blockchain technology continues to evolve, BlockEase stands at the forefront, offering a reliable and scalable solution for the dynamic demands of the digital era. Moving forward, further research and development efforts aimed at refining and expanding the capabilities of BlockEase hold the potential to redefine the landscape of data integrity and security in the modern world. Through collaborative efforts and ongoing innovation, BlockEase paves the way for a future where trust, transparency, and efficiency are the cornerstones of digital transactions and data management.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 07-2024
DOI: 10.5281/zenodo.12743067

Declarations

- **Funding:** We have no direct funding or financial support for this research project. All expenses associated with this study were covered by the authors.
- **Conflict of Interest/Competing Interests:** The authors declare that they have no conflicts of interest or competing interests related to this research.
- **Ethics Approval:** Not Applicable to this paper.
- **Consent to Participate:** Not Applicable to this paper.
- **Consent for Publication:** All authors listed on this manuscript have reviewed and approved the final version for submission and publication.
- **Availability of Data and Materials:** Not Applicable to this paper.
- **Code Availability:** Not Applicable to this paper.
- **Authors' Contributions:** The contributions of each author to this research are as follows:
  - *Ratnesh Kumar Dixit:* Conceived the idea and prepared the draft of the paper.
  - *Nayaneesh Kumar Mishra:* Helped in the experiments and helped in the preparation of the draft of the paper.
  - *R. K. Shukla:* Refined the concept of the paper, reviewed the draft of the paper and gave important feedbacks.
  - *Ratnesh Mishra:* Refined the concept of the paper, reviewed the draft of the paper and gave important feedbacks.
  - *Ravi Shankar Shukla:* Refined the concept of the paper, reviewed the draft of the paper and gave important feedbacks.

References

1) Aitzhan NZ, Svetinovic D (2016) Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. IEEE Transactions on Dependable and Secure Computing 15(5):840–852
2) Aldossary S, Allen W (2016) Data security, privacy, availability and integrity in cloud computing: issues and current solutions. International Journal of Advanced Computer Science and Applications 7(4)
3) Ali M, Khan SU, Vasilakos AV (2015) Security in cloud computing: Opportunities and challenges. Information sciences 305:357–383
4) Alphand O, Amoretti M, Claeys T, et al (2018) Iot chain: A blockchain security architecture for the internet of things. In: 2018 IEEE wireless communications and networking conference (WCNC), IEEE, pp 1–6
5) Croman K, Decker C, Eyal I, et al (2016) on scaling decentralized blockchains: (a position paper). In: International conference on financial cryptography and data security, Springer, pp 106–125
6) Li Z, Kang J, Yu R, et al (2017) Consortium blockchain for secure energy trad- ing in industrial internet of things. IEEE transactions on industrial informatics 14(8):3690–3700
7) Malik A, Nazir MM (2012) Security framework for cloud computing environment: A review. Journal of Emerging Trends in Computing and Information Sciences 3(3):390–394

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 07-2024
DOI: 10.5281/zenodo.12743067

8) Nehe M, Jain SA (2019) A survey on data security using blockchain: Merits, demerits and applications. In: 2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC), pp 1–5, https: //doi.org/10.1109/ICRAECC43874.2019.8995064

9) Puthal D, Malik N, Mohanty SP, et al (2018) everything you wanted to know about the blockchain: Its promise, components, processes, and problems. IEEEConsumer Electronics Magazine 7(4):6–14

10) Rajeswari S, Kalaiselvi R (2017) Survey of data and storage security in cloud computting. In: 2017 IEEE International Conference on Circuits and Systems (ICCS), IEEE, pp 76–81

11) Rao BT, et al (2016) A study on data storage security issues in cloud computing. Procedia Computer Science 92:128–135

12) Sun Y, Zhang J, Xiong Y, et al (2014) Data security and privacy in cloud computing. International Journal of Distributed Sensor Networks 10(7):190903