**Introduction:**

The Idea Concept Paper provides a focused problem direction in the field of Asynchronous Consensus and Permissionless Systems. Asynchronous Permissionless systems are essential for applications operating in decentralized trustless environments because they eliminate the need for a central authority without timing assumptions which enhances security and resilience (Gencer et al., 2018). However, the communication overhead in these systems can become problematic as the network grows, posing a significant barrier to their effectiveness and efficiency. This paper focuses on the improvement of a Reliable Broadcast Communications (RBC) protocol to reduce communication complexity and improve scalability. Specifically, the paper explores how these challenges manifest in the Aleph protocol, one of the first asynchronous consensus protocols to operate in a permissionless setting (Gagol et al., 2019), and discusses strategies to enhance its RBC implementation.

Asynchronous permissionless consensus systems are essential in environments that require decentralized and trustless operations by addressing challenges in security and transparency. In decentralized finance (DeFi), the removal of centralized entities is necessary. By relying on consensus protocols, DeFi platforms ensure that financial transactions are secure and transparent while combating the risks of fraud and manipulation that central entities often introduce (Singh et al., 2022). Similarly, global supply chains benefit from the transparency and trust that decentralization provides. Traditional centralized supply chain systems often have shortcomings such as creating bottlenecks and opacity, but with consensus protocols stakeholders can access an immutable record of transactions that enable trust and seamless operations (Manzoor et al., 2022). In regard to voting systems, where the integrity of the voting process is crucial, consensus systems offer a secure and verifiable means of ensuring that each vote is accurately recorded and immutable. Safeguarding the democratic process (Hajian Berenjestanaki et al., 2024). The growing significance of these applications underscores the importance of asynchronous permissionless systems, which further push the boundaries of what can be achieved in decentralized networks by addressing the blockchain trilemma of scalability, security and decentralization (Principato et al., 2023).

Asynchronous consensus systems are particularly important because they do not require all participants to proceed in lockstep and have no timing assumption. This lack of dependency on synchronized clocks among nodes enhances the robustness of the network against delays and latency (Gao et al., 2022). In contrast, synchronous systems require all nodes to operate within a controlled timing regiment, which can be a significant limitation in decentralized networks where communication delays are unpredictable (Miller et al., 2016).

Aleph is significant because it is one of the first asynchronous consensus protocols to operate in a permissionless setting, eliminating the need for Distributed Key Generation (DKG) and a trusted dealer (Guo et al., 2022). The permissionless nature of Aleph makes it particularly valuable for asynchronous decentralized applications with open participation and no timing assumptions (Guo et al., 2022). However, despite the innovative approach, Aleph's performance is hindered by the communication overhead associated with its Merkle tree-based RBC. As the network scales the overhead becomes an issue and leads to increased resource consumption with reduced efficiency. This is because the Merkle tree structure introduces significant communication overhead as the number of participants grows (Duan et al., 2018). These challenges result in scalability concerns that can affect network performance and resources (Guo et al., 2020). The intention is to address this communication complexity within the protocol's RBC design and implementation with a focus on improving scalability and overall performance.

RBC protocols play a critical role in ensuring the correct and consistent delivery of messages in distributed systems. However, the inherent message complexity in traditional RBC protocols, such as those utilized by HoneyBadgerBFT, Dumbo, and BEAT, presents significant scalability challenges as networks expand (Miller et al., 2016; Duan et al., 2018; Guo et al., 2020). These challenges are particularly pronounced in asynchronous environments, where the growth in communication can lead to bottlenecks, reducing the overall efficiency and effectiveness of the network (Miller et al., 2016; Guo et al., 2020). As the number of participants increases, these limitations highlight the need for more efficient communication mechanisms to maintain robust fault tolerance without sacrificing performance (Duan et al., 2018).

This paper aims to address these challenges within the Aleph protocol. Focusing on refining its RBC implementation to enhance scalability and overall system performance. The goal is to lay the groundwork for future research that explores alternative RBC methods, such as those based on RSA accumulators, with the potential to reduce communication overhead and improve the scalability of asynchronous permissionless systems.

**Problem:**

The communication complexity problem in Aleph’s RBC implementation is highlighted in the Aleph paper on page 32, within the F.1 Lemma, where Aleph's modified version of the RBC protocol, known as Chain Reliable Broadcast (ch-RBC) is discussed (Gągol et al., 2019). This lemma provides a mathematical proof that demonstrates the high message overhead inherent in the Aleph implementation of ch-RBC. The paper elaborates on how the use of Merkle trees in ch-RBC contributes to significant communication overhead and mentions the potential for improvement by converting from Merkle trees to RSA accumulators, as noted in Footnote 26. The F.1 Lemma specifically outlines that ch-RBC has a communication complexity of O(Tr + N² log N), where (Tr) represents the total number of transaction inputs in honest units of round (r), and (N² log N) reflects the communication overhead. This overhead arises from the Propose, Prevote, and Commit phases, which grow quadratically O(N²) due to node-to-node communication, and the Merkle-tree validation phase, which grows logarithmically O(log N) as nodes verify shares using Merkle tree branches (Gągol et al., 2019).

This complexity becomes particularly significant in large networks. For example, with N=1000, the communication complexity results in approximately 9.97 million message exchanges (Gągol et al., 2019). While Aleph makes performance claims, the potential bottleneck posed by this communication complexity remains, especially in permissionless environments where the number of participants can grow rapidly. Addressing this scalability challenge is crucial to enhancing the overall efficiency and performance of the Aleph protocol.

Applications in DeFi and supply chain management need scalable protocols to handle large transaction volumes efficiently and securely. In DeFi, high transaction throughput is essential to prevent bottlenecks and ensure fast processing times while maintaining liquidity. Which in turn supports user trust and market stability. Without scalability, the system could struggle under high demand, leading to delays and reduced confidence (Zhou et al., 2020). Similarly in supply chains, scalable solutions are critical for maintaining transparency and traceability as the network expands. Scalable protocols allow the system to handle increasing data and transaction loads while providing accurate and timely tracking of goods. Preventing fraud and ensuring smooth operations (Manzoor et al., 2022). These examples underscore the necessity for robust and scalable consensus protocols to support the efficiency and growth of modern decentralized applications.

Compared to other RBC protocols, Alephs ch-RBC's O(Tr + N² log N) communication complexity is relatively high especially when compared to permissioned environments. For instance, other RBC techniques such as those based on threshold cryptography or gossip-based dissemination can achieve O(N) or O(N log N ) complexity (Guo et al., 2020, Duan et al., 2018). Permissioned protocols reduce the need for every node to communicate directly with every other node.For example, threshold cryptography requires only a subset of nodes to collaborate for critical operations, minimizing the number of direct interactions needed across the entire network.Similarly, gossip-based dissemination relies on probabilistic communication, where nodes spread information randomly among a small subset of peers.Over multiple rounds the approach ensures that information reaches all nodes without exhaustive communication between each node. By utilizing these communication strategies, these protocols achieve better scalability and lower communication overhead, making them more efficient in handling large-scale networks compared to Aleph’s ch-RBC (Gao et al., 2022).In contrast, the high complexity of Aleph’s ch-RBC becomes a bottleneck in permissionless asynchronous consensus protocols. In these permissionless systems the network size can be large and unpredictable causing the communication overhead to increase significantly as the number of nodes grows.This high communication complexity leads to inefficiencies because each node must manage a larger volume of messages, which can slow down the overall performance of the network.As the number of participants expands the burden of processing these messages grows. Ultimately affecting the system's scalability and responsiveness. This underscores the need for more scalable solutions that can handle large dynamic networks without compromising performance.

The extensive message exchanges required by Aleph’s ch-RBC can lead to network congestion, increased latency, and higher resource consumption. Making it less suitable for large-scale deployment compared to more efficient RBC protocols (Guo et al., 2022). Other permissionless synchronous protocols such as Bitcoin and Ethereum, also encounter scalability issues due to high communication complexity but usually less than their asynchronous counterparts.

For example, Bitcoin’s Nakamoto Consensus has an implicit complexity of O(), while Ethereum’s original Proof-of-Work exhibits similar linear complexity (Gencer et al., 2018). These protocols benefit from their synchronous or semi-synchronous designs, which help manage communication overhead more effectively. Hashgraph, although asynchronous, achieves a communication complexity of O(N log⁡N) and is more practical in semi-synchronous settings due to its efficient gossip and virtual voting mechanisms (Baird & Luykx, 2020). In contrast, Aleph’s ch-RBC protocol, with a complexity of O(Tr + N² log N), faces greater challenges because it is designed for fully asynchronous environments. The higher complexity of Aleph arises from the need for extensive message exchanges among all nodes, making it less efficient than synchronous or semi-synchronous systems in handling scalability (Guo et al., 2020).

**Goal:**

The goal of this research is to reduce the communication complexity of Aleph's protocol by implementing an improved RBC protocol using RSA accumulators instead of Merkle trees. The current Aleph protocol has a communication complexity of O(Tr + N² log N), where the overhead is significant as the network scales. By integrating RSA accumulators, the goal is to lower the communication complexity compared to the original Aleph protocol.

To demonstrate the effectiveness of the new RBC protocol the research will focus on several key metrics. Transaction throughput will be measured to assess scalability improvements by determining the number of transactions processed per second. Latency will be evaluated by analyzing the time taken for transaction confirmation which will help determine if the new protocol reduces delays compared to the existing Merkle tree-based approach. Additionally, communication overhead will be compared between the current and new RBC protocols by quantifying the total number of messages exchanged during the consensus process. Resource utilization will also be analyzed to examine the computational resources required by each protocol.

Simulation methodologies and comparative analysis techniques will be utilized to assess and validate the improvements in communication complexity, ensuring that the proposed enhancements deliver measurable advantages over current solutions. To achieve this the research will replicate development environments and benchmarking methodologies used in the HoneyBadgerBFT (HBFT) protocol, ensuring consistent and reliable performance data (Miller et al., 2016, Knudsen et al., 2021).

For example, HBFT was tested on Amazon EC2 instances across various network sizes. Specifically with 32, 40, 48, 56, 64, and 104 nodes. These diverse network environments allowed researchers to evaluate how HBFT performs in networks of different sizes. Similarly, the research on Aleph will involve testing across these network sizes to ensure relevance and accuracy in performance comparison.

The workload conditions in HBFT experiments varied the batch size of transactions that each node proposed 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, and 131072 transactions per batch. The constant transaction size of 250 bytes was also used. This wide range allowed researchers to assess the impact of different transaction volumes on the protocol's throughput and latency. By testing with varying batch sizes the research aimed to identify optimal conditions and performance bottlenecks. The Aleph research simulations will also employ these varying transaction batch sizes to evaluate how the protocol's modifications impact performance under different workload conditions.

In this research, the Aleph protocol will be evaluated by mirroring the HBFT approach to performance metrics, specifically focusing on throughput and latency. Throughput will be measured as the number of transactions committed per second and latency will be assessed as the time from when the first node receives a client request to when the (N−f)th node completes the consensus protocol. Aleph will undergo testing across similar network sizes and workload conditions as those mentioned above to ensure a valid comparison.

By adopting HBFT's environments the study aims to isolate the effects of switching from Merkle trees to RSA accumulators in the RBC protocol, providing clear evidence of any improvements in communication complexity and scalability without confounding factors.

**Review of the Literature:**

The literature review presents a thorough examination of the research contributions that form the foundation for this study and focus on the challenges and potential solutions related to communication complexity in Aleph’s ch-RBC protocol. A central theme of this review is the communication complexity problem inherent in Aleph, where the F.1 Lemma plays a crucial role. The F.1 Lemma mathematically demonstrates how Aleph’s ch-RBC protocol encounters a high O(Tr + N² log N) complexity. This level of complexity arises due to the extensive message passing and validation requirements across multiple nodes within the network leading to significant overhead. As networks scale in large permissionless environments the messaging overhead exacerbates scalability issues. Making it difficult for Aleph to maintain efficient performance as the number of nodes increases (Gągol et al., 2019).

This detailed analysis underscores the importance of addressing Aleph’s communication overhead as a critical bottleneck that limits the protocol's ability to scale effectively. The need for more efficient solutions becomes evident when considering the growing demands of decentralized applications where high throughput and low latency are essential for maintaining network performance. The review provides context by comparing Aleph with other consensus protocols such as HBFT and Hyperledger Fabric, which have been designed with different approaches to managing communication complexity and enhancing scalability. HBFT for example uses asynchronous byzantine fault tolerance, erasure coding, and threshold cryptography to minimize communication overhead. Allowing HBFT to handle large networks more efficiently (Miller et al., 2016). Hyperledger Fabric on the other hand leverages its modular architecture, ordering services, and gossip protocols to optimize network communication. Often resulting in lower effective communication complexity (Woznica & Kedziora, 2022).

The review goes further by addressing the critical need for enhanced RBC efficiency. Particularly in the context of largely scaled applications such as DeFi and global supply chain management. In these environments the ability to process high transaction volumes in real time is crucial. Protocols like Aleph face significant challenges in meeting these demands primarily due to their existing communication complexity issues. Manzoor et al. (2022) explain how these scalability challenges are common among similar protocols and emphasize the necessity for efficient RBC mechanisms to support the high throughput and low latency required by such applications.

By citing research on other consensus protocols like Dumbo and HBFT, the review establishes that scalability is not just a challenge but also a recognized and actively pursued goal within the field of asynchronous consensus. For example, Dumbo addresses scalability by utilizing several key techniques to minimize communication complexity. It employs an efficient message aggregation mechanism. Which consolidates multiple messages into fewer exchanges, thereby reducing overall communication overhead. Additionally, Dumbo integrates an optimized gossip protocol to facilitate the swift and efficient dissemination of information across the network. This protocol enhances the speed and effectiveness of message sharing further decreasing communication complexity. Furthermore, Dumbo reduces redundancy by focusing on essential communications and minimizing unnecessary exchanges. Collectively, these strategies enable Dumbo to manage larger networks more effectively with lower communication demands, making it a relevant point of comparison for potential improvements to Aleph’s protocol (Duan et al., 2018). HBFT also achieves scalability in asynchronous environments through several key optimizations. Its asynchronous byzantine fault tolerance framework allows nodes to operate independently of synchronized global clocks, effectively managing network delays and faults without extensive message passing. HBFT also employs erasure coding, which divides messages into fragments distributed across nodes, requiring only a subset of fragments to reconstruct the original message and thus reducing overall communication overhead. Additionally, HBFT uses threshold cryptography, where a message is validated by a predefined number of nodes rather than requiring unanimous agreement, further decreasing communication demands. These mechanisms collectively enable HBFT to scale efficiently in large networks by minimizing message passing and managing node failures effectively. Making its research particularly relevant for improving Aleph’s protocol (Miller et al., 2016).

Research supports the proposal to enhance Aleph’s scalability through RSA accumulators. Studies by Reddy (2021) and Hussein & Al-Gailani (2022) provide empirical evidence demonstrating the advantages of RSA accumulators in reducing verification overhead. Reddy (2021) provides a detailed analysis of RSA accumulators and their impact on reducing verification overhead in consensus protocols. In the study empirical tests demonstrated that using RSA accumulators reduced the verification time of consensus protocols from an average of 200 milliseconds per transaction to just 10 milliseconds per transaction. This reduction in verification time was attributed to the efficiency of RSA accumulators in compressing multiple transactions into a single compact proof. Additionally, the study showed that the network bandwidth required for consensus operations decreased by approximately 90%, from 1 MB per block to 100 KB per block, due to the smaller size of the RSA proofs compared to the full transaction set. Additionally, Hussein & Al-Gailani (2022) provides further empirical data showing that RSA accumulators significantly improve system performance by reducing verification time and data size in blockchain networks. For example, their simulations demonstrated that for 50 transactions, RSA accumulators with a 128-bit setup reduced the data size required for verification to approximately 3,977 bytes, compared to Merkle tree-SHA256 and Merkle tree-BLAKE256 which required significantly larger sizes, around 17.6 KB for similar setups. Additionally, the running time for verification using RSA accumulators was significantly lower, with the accumulator setup taking only 60 ms compared to the Merkle tree methods which took between 2,500 ms and 5,460 ms. These results highlight the efficiency gains achieved with RSA accumulators especially in handling large transaction volumes.

While RSA accumulators have been primarily tested in synchronous and permissioned environments. Their attributes such as reducing verification overhead and handling large datasets more efficiently suggest promising potential for asynchronous permissionless settings like Aleph. Given their demonstrated benefits in other contexts, exploring RSA accumulators in asynchronous permissionless environments could offer significant scalability improvements for Aleph and address key performance challenges identified in previous research.

This literature review substantiates the proposed approach by demonstrating how existing research aligns with the study's goals. The use of RSA accumulators supported by evidence from previous research offers a viable solution to the communication complexity and scalability challenges identified in Aleph. The comprehensive analysis provided in this review not only highlights the limitations of Aleph’s current implementation but also points towards practical strategies for enhancing its performance in large scale decentralized networks thus laying a solid foundation for the research's proposed enhancements.

**Approach**

To achieve the goal of reducing the communication complexity of Aleph by implementing RSA accumulators in place of Merkle trees in the RBC protocol the research will follow a structured approach. This method ensures a comprehensive evaluation of the proposed improvements across various experimental environments focusing on key metrics that reflect the protocol's performance and scalability.

The technical implementation of the research will be written in Rust, leveraging the Tokio asynchronous runtime for efficient and concurrent network communication, as used in the Asynchronous Byzantine Fault Tolerant (ABFT) protocols research (Knudsen et al., 2021). This setup will be deployed on Amazon Web Services (AWS) or a school provided resource, utilizing a combination of Ansible for configuration management and AWS Cloud Development Kit (AWS CDK) for infrastructure provisioning. This infrastructure will support thorough testing and evaluation of the original and modified Aleph protocol under realistic network conditions.

The baseline for this research will be the original Aleph protocol, which utilizes Merkle trees in its RBC. This version will be benchmarked to capture current metrics such as communication complexity, transaction throughput, latency, and resource utilization. The proposed modification involves replacing Merkle trees with RSA accumulators in the RBC protocol. This change will incorporate the RSA accumulator-based aggregation and verification process, streamline message passing, and reduce the overhead associated with Merkle trees.

The implementation details of this research will replace the Merkle tree structure in the current ch-RBC protocol with RSA accumulators. These accumulators will aggregate transaction data into a single, compact representation, aiming to reduce message complexity from O(Tr + N² log N) to a lower complexity. RSA accumulators will also simplify the verification process, allowing nodes to confirm the integrity and validity of broadcasted messages with reduced computational effort.

The modified Aleph protocol will be assessed in an experimental environment that mirrors the HBFT setup to ensure consistency and valid comparisons. Each test will use the transaction size of 250 bytes per batch as used in HBFT. The first test will use 32 nodes with a batch size of 256 transactions. Subsequent experiments will incrementally the node count to 40, 48, 56, 64, and up to 104 nodes, with batch sizes of 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, and 131072 transactions. Contingent on available resources. This systematic approach will evaluate the scalability of the modified Aleph protocol by assessing how varying node counts and batch sizes impact throughput and latency, helping to identify optimal conditions and potential bottlenecks.

Metrics for evaluation will include transaction throughput, which is measured as the number of transactions processed per second to determine if the modified protocol offers scalability improvements by handling larger transaction volumes more efficiently. Latency will be assessed as the time taken for transaction confirmation specifically from when the first node receives a client request to when the (N−f)th node completes the consensus process. Communication overhead will be quantified by the total number of messages exchanged during the consensus process to compare the communication complexity between the current and new RBC protocols. Resource utilization will be analyzed in terms of CPU and memory usage across nodes during the RBC process.

For data collection, real-time logging will be handled using the tracing framework in Rust, which supports structured logging and is well-suited for asynchronous code. Key metrics such as transaction throughput, latency, communication overhead, and resource utilization will be recorded and securely logged during each experiment. The data will be stored in PostgreSQL, ensuring reliable and efficient management of the collected information. This approach will facilitate easy access and analysis of the metrics supporting a thorough evaluation of the modified Aleph protocol.

For comparative analysis, Jupyter Notebooks will be utilized to gather and display information. After completing the experiments data will be extracted and analyzed within the notebook environment. This setup allows for interactive exploration of the data, using statistical methods to compare the performance of the original and modified systems. Jupyter Notebooks support rich text documentation and inline visualizations making it easy to present results through comparative graphs and statistical summaries. This approach will effectively illustrate the impact of RSA accumulators on Aleph’s communication complexity and scalability.

**References:**

Amazon Web Services, Inc. (n.d.). *AWS Cloud Development Kit (AWS CDK)*. Retrieved August 26, 2024, from<https://aws.amazon.com/cdk/>

Baird, L., & Luykx, A. (2020). The Hashgraph protocol: Efficient asynchronous BFT for high-throughput distributed ledgers. In *International Conference on Omni-layer Intelligent Systems* (pp. 1-7).<https://doi.org/10.1109/COINS49042.2020.9191430>

Djari, A., Anceaume, E., & Tucci-Piergiovanni, S. (2022). An extensive agent-based simulation study of sycomore++, a DAG-based permissionless ledger. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing* (pp. 334–336). Association for Computing Machinery.<https://doi.org/10.1145/3477314.3507245>

Duan, S., Reiter, M., & Zhang, H. (2018). Beat: Asynchronous BFT made practical. In *ACM SIGSAC Conference on Computer and Communications Security* (pp. 2028–2041).<https://doi.org/10.1145/3243734.3243812>

Gągol, A., Leśniak, D., Straszak, D., & Świętek, M. (2019). Aleph: Efficient atomic broadcast in asynchronous networks with Byzantine nodes. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies (AFT '19)* (pp. 214–228). Association for Computing Machinery.<https://doi.org/10.1145/3318041.3355467>

Gao, Y., Lu, Y., Lu, Z., Tang, Q., Xu, J., & Zhang, Z. (2022). Dumbo-NG: Fast asynchronous BFT consensus with throughput-oblivious latency. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22)* (pp. 1187–1201). Association for Computing Machinery.<https://doi.org/10.1145/3548606.3559379>

Gencer, A. E., Basu, S., Eyal, I., van Renesse, R., & Sirer, E. G. (2018). Decentralization in Bitcoin and Ethereum networks. In *Proceedings of the 22nd International Conference on Financial Cryptography and Data Security* (pp. 439-457). Springer.<https://doi.org/10.48550/arXiv.1801.03998>

Guo, B., Lu, Z., Tang, Q., Xu, J., & Zhang, Z. (2020). Dumbo: Faster asynchronous BFT protocols. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)* (pp. 803–818). Association for Computing Machinery.<https://doi.org/10.1145/3372297.3417262>

Guo, B., Lu, Y., Lu, Z., Tang, Q., Xu, J., & Zhang, Z. (2022). Speeding Dumbo: Pushing asynchronous BFT closer to practice.<https://doi.org/10.14722/ndss.2022.24385>

Han, J., Liu, X., & Li, K. (2023). Swarm: A high-performance asynchronous BFT protocol adapted to high network delay. In *Proceedings of the 2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS)* (pp. 1801-1808). Ocean Flower Island, China.<https://doi.org/10.1109/ICPADS60453.2023.00249>

Hajian Berenjestanaki, M., Barzegar, H. R., El Ioini, N., & Pahl, C. (2024). Blockchain-based e-voting systems: A technology review. *Electronics, 13*(1), 17.<https://doi.org/10.3390/electronics13010017>

Hussein, K. M., & Al-Gailani, M. F. (2022). An efficient bandwidth based on the cryptographic technique of the RSA accumulator in blockchain networks. In *2022 Fifth College of Science International Conference of Recent Trends in Information Technology (CSCTIT)* (pp. 164-168). Baghdad, Iraq.<https://doi.org/10.1109/CSCTIT56299.2022.10145614>

Jupyter. (n.d.). Jupyter notebooks. Retrieved August 26, 2024, from https://jupyter.org/

Kharangate, A. (2023). Asynchronous Merkle trees. arXiv.<https://ar5iv.labs.arxiv.org/html/2311.17441>

Knudsen, H., Li, J., Notland, J., Haro, P., & Ræder, T. (2021). High-performance asynchronous Byzantine fault tolerance consensus protocol. In *IEEE International Conference on Blockchain* (pp. 476-483).<https://doi.org/10.1109/Blockchain53845.2021.00073>

Lauinger, J., Ernstberger, J., Regnath, E., Hamad, M., & Steinhorst, S. (2021). A-PoA: Anonymous proof of authorization for decentralized identity management. In *2021 IEEE International Conference on Blockchain and Cryptocurrency* (ICBC) (pp. 1-9). Sydney, Australia. doi:10.1109/ICBC51069.2021.9461082

Manzoor, R., Sahay, B. S., & Singh, S. K. (2022). Blockchain technology in supply chain management: An organizational theoretic overview and research agenda. *Annals of Operations Research*.<https://doi.org/10.1007/s10479-022-05069-5>

Miller, A., Xia, Y., Croman, K., Shi, E., & Song, D. (2016). The honey badger of BFT protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 31–42).<https://doi.org/10.1145/2976749.2978399>

PostgreSQL. (n.d.). *PostgreSQL: The world's most advanced open source relational database*. Retrieved August 26, 2024, from<https://www.postgresql.org/>

Principato, M., Babel, M., Guggenberger, T., Kropp, J., & Mertel, S. (2023). *Towards solving the blockchain trilemma: An exploration of zero-knowledge proofs*. In *ICIS 2023 Proceedings* (pp. 1-15). Association for Information Systems.<https://aisel.aisnet.org/icis2023/blockchain/blockchain/5>

Reddy, B. S. (2021). securePrune: Secure block pruning in UTXO based blockchains using accumulators. In *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)* (pp. 174-178). Bangalore, India.<https://doi.org/10.1109/COMSNETS51098.2021.9352892>

The Rust Project Developers. (n.d.). *The Rust programming language*. Retrieved August 26, 2024, from<https://www.rust-lang.org/>

Singh, C., Thakkar, R., & Warraich, J. (2022). Blockchain in supply chain management. *European Journal of Engineering and Technology Research, 7*(5), 60-69.<https://doi.org/10.24018/ejeng.2022.7.5.2888>

The Tracing Project. (n.d.). *tracing: Documentation*. Retrieved August 26, 2024, from<https://docs.rs/tracing/latest/tracing/>

Tokio Contributors. (n.d.). *Tokio*. Retrieved August 26, 2024, from<https://tokio.rs/>

Woznica, A., & Kedziora, M. (2022). Performance and scalability evaluation of a permissioned blockchain based on Hyperledger Fabric, Sawtooth and Iroha. *Comput. Sci. Inf. Syst., 19*, 659-678.

Zhou, Q., Huang, H., Zheng, Z., & Bian, J. (2020). Solutions to scalability of blockchain: A survey. *IEEE Access*. Advance online publication.<https://doi.org/10.1109/ACCESS.2020.2967218>