Implementing RSA Accumulators for Asynchronous and Permissionless Reliable Broadcasting

Eric Webb

A dissertation idea paper submitted in partial fulfillment of the requirements of the degree of Doctor of Philosophy in Cybersecurity Management

College of Computing and Engineering, Nova Southeastern University

1-11-2025

**Introduction**

The presented research focuses on asynchronous consensus and permissionless systems that eliminate the need for centralized authority and timing assumptions. These systems improve resilience and security in applications and thus are essential for decentralized trustless environments such as decentralized finance (DeFi), supply chain management, and voting systems (Hajian Berenjestanaki et al., 2024). However as these decentralized networks expand the communication overhead can increase significantly and lead to performance bottlenecks.

The Aleph permissionless asynchronous protocol’s implementation of a Reliable Broadcasting (RBC) protocol called Chain Reliable Broadcast (ch-RBC), degrades due to the communication complexity that increases quadratically with the number of nodes. Thus to address this performance degradation an enhancement is proposed in this work by integrating RSA accumulators into Alephs ch-RBC to reduce communication complexity and improve scalability.

Decentralized systems in asynchronous and permissionless environments face significant scalability challenges particularly due to the communication complexity inherent in RBC protocols, which are vital for achieving consensus. Traditional RBC designs often based on Merkle trees for transaction validation exhibit quadratic communication complexity as the number of participants grows (Miller et al., 2016; Gągol et al., 2019). This inefficiency places substantial strain on network resources such as bandwidth and computational capacity, which hinders performance and scalability.

These challenges are especially pronounced in critical applications. In DeFi for example, high transaction throughput and robust security are essential to maintaining liquidity and trust (Singh et al., 2022). Similarly, traditional centralized supply chain systems often suffer from bottlenecks and opacity. In contrast, consensus protocols enable stakeholders to access immutable and transparent transaction records that foster trust and operational efficiency (Manzoor et al., 2022). In voting systems, where integrity and accuracy are paramount, consensus mechanisms ensure each vote is securely and verifiably recorded such that it safeguards the electoral process (Hajian Berenjestanaki et al., 2024).

The importance of addressing these limitations highlights the value of asynchronous permissionless systems. By tackling the blockchain trilemma of balancing scalability, security, and decentralization these systems enable greater efficiency and robustness in real-world applications (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. The 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 and can be a significant limitation in decentralized networks where communication delays are unpredictable (Miller et al., 2016).

Aleph is one of the first asynchronous consensus protocols to operate in a permissionless setting and eliminates the need for Distributed Key Generation (DKG) and a trusted dealer (Guo et al., 2022). Its innovative ch-RBC protocol enables decentralized operations with open participation and no timing assumptions making it a valuable solution for asynchronous decentralized applications. However Aleph’s reliance on Merkle trees introduces significant communication overhead as the network scales. The quadratic complexity of message verification and broadcasting results in higher resource consumption and reduced efficiency. These scalability concerns limit its performance in larger networks where robust and efficient operations are critical. Addressing these challenges are essential to improving Aleph’s scalability and overall effectiveness in asynchronous and permissionless environments.

RBC protocols play a critical role in ensuring the correct and consistent delivery of messages in distributed systems. However, as 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 and reduce 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).

The research focus in this paper is the integration of RSA accumulators into Aleph's ch-RBC protocol to address the scalability and communication overhead challenges inherent in its Merkle tree-based design. By reducing the communication complexity this approach aims to enhance the protocol's performance and scalability in asynchronous and permissionless environments. The research contributes to the broader goal of improving decentralized networks by offering a more efficient consensus mechanism, lowering resource consumption, and maintaining robustness. These improvements are particularly valuable for applications in DeFi, supply chain management, and secure voting systems which demand high efficiency, transparency, and fault tolerance.

**Problem**

Aleph’s ch-RBC protocol is a critical component of its asynchronous and permissionless consensus layer but suffers from significant communication complexity as network size increases. This limitation is detailed in the Aleph paper where the authors demonstrate that the protocol's communication complexity scales as O(Tr + N² log N), where Tr represents the total number of transactions and N is the number of participating nodes (Gągol et al., 2019). This complexity stems from the use of Merkle trees that ensure consistency and fault tolerance but imposes substantial overhead due to the logarithmic growth in validation costs and the quadratic growth in node-to-node message exchanges.

As the number of participants approaches 1,000 nodes the communication overhead in Aleph’s ch-RBC escalates dramatically reaching nearly 10 million message exchanges. This overhead arises primarily during the propose, prevote, and commit phases which requires each node to communicate with every other node multiple times. Such intensive messaging places a heavy burden on network bandwidth and node processing capabilities creating bottlenecks that degrade performance. These challenges are especially detrimental in permissionless environments where the number of nodes can grow unpredictably (Gągol et al., 2019; Duan et al., 2018).

Aleph’s high communication complexity not only affects throughput but also limits scalability making the protocol less suitable for real-world applications such as decentralized finance (DeFi) and supply chain management. Both domains demand consensus protocols capable of handling large transaction volumes while ensuring performance and reliability. For instance, DeFi platforms require high throughput to maintain liquidity and prevent market instability, while supply chain networks rely on transparency and efficiency to track and verify goods across multiple stakeholders (Zhou et al., 2020; Manzoor et al., 2022). In Aleph’s current state the communication overhead hampers its ability to scale effectively in such scenarios.

When compared to other consensus protocols, Aleph’s O(Tr + N² log N) complexity is particularly high. For example, protocols employing threshold cryptography or gossip-based dissemination achieve lower complexities such as O(N²) (Guo et al., 2020). These techniques reduce the need for direct communication between all nodes and instead rely on smaller subsets of nodes or probabilistic communication strategies to disseminate information. In contrast, Aleph’s design necessitates extensive message exchanges among all participants and leads to inefficiencies in large-scale networks.

Even among permissioned asynchronous protocols Aleph’s communication overhead is notable. HoneyBadgerBFT and Dumbo face similar challenges with quadratic message complexities resulting in bottlenecks as network size increases (Miller et al., 2016; Duan et al., 2018). While synchronous or semi-synchronous protocols such as Bitcoin’s Nakamoto Consensus and Hashgraph exhibit lower complexities O(N) and O(N log N) respectively. These systems benefit from their less stringent assumptions and controlled environments (Gencer et al., 2018; Baird & Luykx, 2020). In contrast, Aleph’s fully asynchronous and permissionless design makes it inherently more complex and challenging to optimize (Guo et al., 2020).

The extensive message exchanges required by Aleph’s ch-RBC result in network congestion, increased latency, and higher resource consumption, making it unsuitable for large-scale deployment. Addressing this communication complexity is crucial to improving the protocol’s scalability and ensuring its viability for real-world decentralized applications.

**Goal**

The research aims to implement a RBC protocol that incorporates RSA accumulators as a replacement for the Merkle trees currently used in Aleph’s ch-RBC. Aleph’s current protocol has a communication complexity of O(Tr + N² log N), where Tr represents the total number of transactions and N is the number of participating nodes. By integrating RSA accumulators, the goal is to reduce this complexity to O(Tr + N²) thereby addressing the scalability and efficiency challenges inherent in the existing design. Prior studies have shown that RSA accumulators are effective in reducing message complexity and validation overhead making them a promising approach to improving the performance of Aleph’s ch-RBC protocol in permissionless networks (Gągol et al., 2019; Hussein & Al-Gailani, 2022; Reddy, 2021).

To validate the effectiveness of the RSA-based RBC protocol the research will measure key performance metrics including throughput, latency, communication overhead, and resource utilization. Throughput will be used to evaluate the protocol’s capacity to handle large transaction volumes by measuring the number of transactions processed per second. Latency will provide insight into the system’s responsiveness by assessing the time taken from transaction submission to its finalization. Communication overhead will be analyzed by quantifying the total number of messages exchanged during the consensus process which reflects the efficiency of the protocol’s design. Resource utilization will include CPU and memory consumption, and will help determine the protocol’s ability to operate efficiently under varying workloads. These metrics are standard in asynchronous consensus protocol evaluations and have been used in studies of HoneyBadgerBFT, BEAT, and Dumbo (Miller et al., 2016; Duan et al., 2018; Guo et al., 2020).

Scalability will be demonstrated by observing how these metrics behave as the number of nodes and transaction volumes increase. A successful reduction in communication overhead with improved throughput and latency in larger networks will illustrate the scalability of the proposed approach. This analysis will provide concrete evidence of the impact of RSA accumulators on the protocol’s performance.

The research involves implementing Aleph’s ch-RBC proof in a controlled programming environment and replacing the Merkle tree structure with an RSA accumulator-based design. This process includes developing both the Merkle tree-based control protocol and the RSA accumulator-based protocol and comparing their performance in identical simulated environments. To ensure reliability the study will follow established methodologies for benchmarking asynchronous consensus protocols including HoneyBadgerBFT, Dumbo, and ABFT and will simulate diverse network conditions and workloads to evaluate performance under realistic decentralized environments.

By isolating the effects of RSA accumulators on communication complexity the study aims to provide a detailed performance comparison between the two designs. The findings will demonstrate how RSA accumulators can significantly enhance scalability and overall performance in permissionless asynchronous networks. This improvement addresses the pressing issue of communication overhead in Aleph’s ch-RBC protocol ensuring its suitability for real-world decentralized applications in areas such as decentralized finance, supply chain management, and other domains requiring high scalability, transparency, and efficiency.

**Relevance, Significance, and Review of the Literature**

Asynchronous consensus protocols such as HoneyBadgerBFT (Miller et al., 2016), Dumbo (Duan et al., 2018), and Aleph (Gągol et al., 2019), represent significant advancements in decentralized networks. Particularly in their ability to handle unpredictable network delays by operating without timing assumptions. These protocols are crucial in asynchronous environments where participants do not need to proceed in lockstep and enable higher resilience to latency issues and network faults. However, while they provide robust solutions one of the major challenges they face in large-scale networks are the complexity and volume of communication required to maintain consensus.

HoneyBadgerBFT stands as one of the earliest practical asynchronous Byzantine Fault Tolerant (BFT) protocols and is designed to tolerate faulty or malicious nodes while ensuring secure consensus. HBFT's communication complexity scales quadratically O(N²) as the network grows making it less efficient in handling large-scale systems. The quadratic message growth arises from the need for every node to communicate directly with others during consensus contributing to scalability limitations (Miller et al., 2016). Similarly, Dumbo and BEAT protocols improved certain aspects of asynchronous communication efficiency in permissioned environments but both still exhibited significant message complexity especially in networks that aim for a high degree of decentralization (Duan et al., 2018; Guo et al., 2020).

The Aleph protocol introduced a unique take on asynchronous permissionless consensus by avoiding the need for Distributed Key Generation (DKG) and a trusted dealer. Making it an attractive choice for permissionless decentralized applications with open participation. However, like its predecessors Aleph's ch-RBC protocol suffers from high communication complexity, with a performance that scales as O(Tr + N² log N). The complexity comes from the combined effects of node-to-node communication and the use of Merkle trees for transaction validation which contributes to a logarithmic factor in the message complexity (Gągol et al., 2019).

Merkle trees have long been employed for transaction validation in distributed systems due to their ability to offer verifiable proofs of inclusion with logarithmic efficiency. However as networks expand and the number of participants increases, Merkle tree-based validation introduces overhead that becomes unmanageable. The overhead is particularly true in Alephs’s ch-RBC protocol where every transaction requires generating and verifying a Merkle tree proof contributing to the overall message complexity.

In response to these scalability concerns researchers have proposed using RSA accumulators as a more efficient alternative for generating verifiable proofs. RSA accumulators were first introduced by Benaloh and de Mare (1993) and allow for constant-sized proofs regardless of the number of transactions or participants and significantly reduce the communication overhead associated with validation. Empirical studies such as those by Reddy (2021) and Hussein & Al-Gailani (2022) demonstrated the practical benefits of RSA accumulators in reducing both verification time and bandwidth usage in permissioned and synchronous systems. In particular Reddy's work showed that RSA-based verification reduced transaction verification time from 200 milliseconds to just 10 milliseconds and decreased bandwidth requirements by 90%. Similarly Hussein & Al-Gailani's study found that RSA accumulators reduced data size for verifying 50 transactions to 3,977 bytes, compared to over 17.6 KB using Merkle tree-based approaches (Reddy, 2021; Hussein & Al-Gailani, 2022).

In the context of consensus protocols RSA accumulators have the potential to address the specific communication complexity challenges faced by Aleph's ch-RBC. By replacing Merkle trees with RSA accumulators, the Aleph protocol could see a reduction in communication complexity from O(Tr + N² log N) to O(Tr + N²) allowing it to scale more efficiently without sacrificing security or correctness. The shift could lead to significant improvements in both throughput and latency as the network will be able to handle larger transaction volumes with reduced message overhead.

The need for scalable asynchronous permissionless consensus protocols is especially pressing in real-world applications such as decentralized finance (DeFi), global supply chains, and voting systems. In DeFi, high transaction throughput is essential to maintain liquidity and prevent bottlenecks during periods of high demand. Similarly, supply chains require scalability to handle the increasing number of transactions and participants while ensuring transparency and traceability. Scalable consensus protocols are crucial in enabling these systems to grow without suffering from performance degradation (Singh et al., 2022; Manzoor et al., 2022). Voting systems also rely on scalable consensus protocols to ensure the integrity of elections, especially in large-scale elections where the number of participants can fluctuate dramatically (Hajian Berenjestanaki et al., 2024).

In comparison to other RBC protocols Aleph's current communication complexity is notably higher making it less efficient in large decentralized networks. Alternative approaches, such as those used in BEAT or gossip-based protocols, offer lower communication complexity and greater scalability by minimizing the number of direct communications required between nodes (Guo et al., 2020; Gao et al., 2022). These methods focus on reducing the number of messages exchanged during consensus by relying on a subset of nodes for critical operations and thereby limiting the communication overhead. In contrast, Aleph's ch-RBC involves every node in direct communication due to its permissionless nature, leading to increased complexity as the network expands.

The proposed use of RSA accumulators in the research aims to address these challenges by reducing the communication complexity in Aleph's ch-RBC enabling it to scale more effectively in permissionless environments. Through the integration of RSA accumulators the research seeks to demonstrate that Aleph can achieve the same level of fault tolerance and security while significantly improving its scalability and performance.

**Approach**

The research seeks to evaluate the impact of replacing the Merkle trees in Aleph’s ch-RBC protocol with RSA accumulators to address the critical issue of communication complexity. Aleph’s current protocol has a communication complexity of O(Tr + N² log N), where Tr represents the total number of transactions and N is the number of nodes. By replacing the Merkle tree validation mechanism with RSA accumulators, the research aims to reduce this complexity to O(Tr + N²) thereby improving the protocol’s scalability and overall performance in permissionless asynchronous networks (Gągol et al., 2019).

The research will first establish a control system by implementing the Aleph protocol’s Merkle tree-based ch-RBC in Rust. Rust is selected due to its high performance and robust ecosystem for cryptographic implementations which aligns with the requirements of consensus protocols (Knudsen, 2021). The design will adhere to the specifications provided in the Aleph papers RBC protocol (Gągol et al., 2019) ensuring the baseline implementation accurately reflects the original protocol. The ch-RBC proof will serve as the foundation with the Merkle tree structure being implemented for transaction validation as described in lines 3 and 19 of the ch-RBC proof of Aleph. This control system will serve as the benchmark for comparison with the modified RSA-based protocol.

The RSA-based modification will replace the Merkle tree with an RSA accumulator for transaction validation. RSA accumulators are cryptographic constructs that enable compact proofs which significantly reduce communication overhead by allowing validators to verify a batch of transactions using a single accumulator proof (Reddy, 2021). This modification will be made by adjusting the protocol’s transaction validation steps, specifically in the propose, prevote, and commit phases. The RSA accumulator will replace the Merkle tree root by generating a single proof for the transaction shares in each round, leveraging an RSA modulus N and secure prime factors p and q. The accumulator witnesses for each transaction share will be computed and verified during the prevote and commit phases, ensuring the validation process aligns with the protocol’s requirements while reducing the communication overhead.

Validation in this context refers to ensuring that all transactions in a batch are legitimate and correctly included in the protocol’s state. The Merkle tree achieves this by constructing a hash-based tree with the transaction shares as leaves requiring nodes to exchange and validate Merkle proofs. RSA accumulators simplify this by compressing the transaction proofs into a single constant-sized proof, which is verified using modular arithmetic. The approach is grounded in prior research which has demonstrated the efficiency of accumulators in reducing communication complexity while maintaining integrity and correctness (Reddy, 2021; Hussein & Al-Gailani, 2022).

The experimental design involves implementing both versions of the protocol as in the Merkle-tree based and RSA-based RBC using a standardized transaction size of 250 bytes per batch which is consistent with methodologies in prior studies like the HoneyBadgerBFT (Miller et al., 2016). Each node in the network will be represented as an independent AWS EC2 instance and simulate a decentralized environment. The tests will begin with a network of 32 nodes and a batch size of 256 transactions progressively increasing the node count to 104 and the batch size to 131,072 transactions. This range aligns with established benchmarks of HBFT and ABFT (Miller et al., 2016, Knudsen, 2021) and allows for a comprehensive evaluation of scalability and performance under varying network conditions.

The key metrics for evaluation include throughput, latency, communication overhead, and resource utilization. Throughput measured as transactions per second reflects the protocol’s capacity to handle high transaction volumes. Latency measures the time taken from transaction submission to finalization providing insight into system responsiveness. Communication overhead quantifies the total number of messages exchanged during the consensus process and resource utilization captures CPU and memory usage across nodes. These metrics are critical for assessing both efficiency and scalability as prior research has shown that these indicators correlate strongly with a protocol’s ability to perform in large-scale networks (Miller et al., 2016; Gągol et al., 2019).

Logs will be collected using Rust’s logging framework with data stored in an AWS S3 bucket for centralized access. Metrics such as CPU and memory usage will be monitored using CloudWatch to provide real-time insights into resource consumption. The use of AWS infrastructure ensures a controlled and scalable testing environment where network configurations and transaction volumes can be adjusted to simulate real-world conditions.

The collected data will be analyzed to compare the performance of the Merkle tree-based and RSA-based implementations. A PostgreSQL database will be used to aggregate and process the data allowing for detailed analysis of the differences between the two designs. Metrics will be evaluated to determine whether RSA accumulators provide measurable improvements in throughput, latency, and communication overhead, indicating enhanced scalability. The findings will be presented through visualizations generated using Jupyter Notebooks ensuring clarity in the comparison and interpretation of results.

The research anticipates that replacing Merkle trees with RSA accumulators will reduce communication overhead by compressing transaction proofs into constant-sized accumulator proofs. This reduction is expected to decrease the overall message complexity improving throughput and reducing latency. Particularly in networks with a high number of nodes. The experimental results will demonstrate whether the RSA-based design can handle larger transaction volumes and node counts without significant performance degradation. These insights will contribute to understanding how cryptographic constructs like RSA accumulators can enhance the scalability of asynchronous consensus protocols and address the challenges of decentralized systems.

**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>

Cardinal Cryptography. (n.d.). *AlephBFT* [GitHub repository]. GitHub. Retrieved October 14,

2024, from<https://github.com/Cardinal-Cryptography/AlephBFT/tree/main>

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>