**Overview:**

**Introduction:**

The Idea Concept Paper will identify and focus a research direction in the field of Asynchronous Consensus and Permissionless Systems. Specifically, the research will incorporate the Aleph protocol (Gągol et al., 2019) as the basis of the consensus protocol, which aims to improve efficiency and scalability by reducing communication complexity and advancing the general field of Asynchronous Consensus in Permissionless systems. The paper will address a performance problem with Alephs communication complexity and propose an improvement based on the evaluation and integration of published improved Reliable Broadcast Communications (RBC), replacing Aleph’s merkle tree based RBC with one based on RSA accumulators (Hussein & Al-Gailani, 2022).

The Aleph protocol is important because it is credited to being one of the first asynchronous consensus protocols to operate in a permissionless setting by removing the need of Distributed Key Generation (DKG) and operating without a trusted dealer (Guo et al., 2022). However, performance is throttled by the use of merkle tree based RBC, which becomes resource intensive as the network scales. This is because merkle trees provide cryptographic proofs of data integrity, but require significant computational resources for construction and verification leading to increased latency and reduced throughput in blockchain systems. (Hussein & Al-Gailani, 2022).

To address these issues, this research proposes replacing Aleph's merkle tree based RBC with a more efficient system based on RSA accumulators. RSA accumulators offer a cryptographic alternative that can significantly reduce communication and computational complexity (Reddy, 2021). By integrating RSA accumulator based RBC, the protocol aims to decrease communication complexity and lead to an enhanced performance.

Since there is limited research available in the field of Asynchronous Permissionless systems, any enhancement in Aleph’s RBC would not only benefit Aleph itself but also contribute to the broader domain of Asynchronous Permissionless systems. As Aleph is one of the few Asynchronous Permissionless protocols, advancements in its RBC could set a precedent and provide valuable insights for future developments in the domain of Asynchronous Permissionless consensus.

**Objectives**

**Redeploy Original Aleph:**

* **Objective:** Assess the current performance of the Aleph protocol with its existing Merkle Tree-based RBC in various network conditions.
* **Measurement:** Analyze communication complexity, transaction throughput, network latency, and resource utilization under different scenarios.

**Integrate RSA Accumulators:**

* **Objective:** Implement RSA accumulator-based RBC within the Aleph protocol and measure the improvements in communication complexity, latency, and throughput.
* **Measurement:** Quantify the reduction in communication rounds, increase in transaction throughput, and decrease in network latency and resource utilization.

**Compare and Contrast:**

* **Objective:** Conduct a comparative analysis of the Merkle Tree-based and RSA accumulator-based RBC implementations to quantify the effectiveness of the proposed improvement.
* **Measurement:** Perform side-by-side comparisons of performance metrics such as communication complexity, throughput, latency, resource consumption, and communication rounds.

**Scalability Testing:**

* **Objective:** Test the scalability of both the original and the improved Aleph protocol in large-scale network simulations to ensure it can handle increased node participation without significant performance degradation.
* **Measurement:** Evaluate the protocol's performance with varying numbers of nodes and transaction loads, and monitor for any signs of bottlenecks or performance issues.

**Resource Projections:**

**Infrastructure (AWS or school provided)**

* Compute: Utilize AWS EC2 instances for hosting network nodes, consensus algorithms, and other protocol components.
* Storage: Employ AWS S3 buckets or EBS volumes for storing blockchain ledger data, transaction history, and protocol metadata.
* Networking: Utilize AWS VPCs, Route 53 for DNS resolution, and other networking services to manage communication between network nodes.

**Third-Party Libraries or Services for Bandwidth Control:**

* Implement bandwidth control within the protocol codebase using third-party libraries and services such as Apache Traffic Server (ATS), Nginx with Traffic Shaping Module, or libraries like tokio-ratelimit for Rust.
* Utilize AWS services like Amazon CloudFront or AWS Global Accelerator for content delivery and bandwidth optimization.

**Monitoring and Management Tools:**

* Implement monitoring and management tools such as AWS CloudWatch, AWS Config, and AWS Systems Manager for monitoring resource utilization, performance metrics, and automated management of deployed resources.
* Use third-party monitoring solutions for additional insights into network health, performance, and security.

**Automation with Ansible and AWS CDK**

* **Infrastructure as Code:** Define and deploy AWS infrastructure using AWS CDK, enabling version control and reproducibility.
* **Resource Management:** Use AWS CDK to model and provision compute resources (EC2 instances), storage solutions (S3, EBS), and networking configurations (VPCs, Route 53).
* **Deployment Automation:** Use Ansible playbooks to automate the provisioning and configuration of AWS EC2 instances, VPCs, S3 buckets, and other infrastructure components.
* **Configuration Management:** Manage the installation and configuration of third-party libraries for bandwidth control, monitoring tools, and security settings using Ansible.

**Milestones and Deliverables**

**Scheduled completions**

**Milestone 1: Approval for Dissertation Research (ISEC885)**

Scheduled Completion: [2024]

Deliverable:

* Approval documentation and research proposal outlining the objectives, methodology, and significance of the research project.

**Milestone 2: Redeploy Aleph on AWS and Measure (ISEC901 - 1)**

Scheduled Completion: [2025]

Deliverable:

* Redeployment of Aleph protocol on AWS infrastructure.
* Measurement of communication complexity and performance metrics or the original Aleph protocol.

**Milestone 3: Redeploy Aleph with RSA Accumulators and Measure (ISEC901 - 2)**

Scheduled Completion: [2026]

Deliverable:

* Implementation of RSA accumulators into the Aleph protocol.
* Redeployment of the modified Aleph protocol on AWS infrastructure.
* Measurement of communication complexity and performance metrics of the modified Aleph protocol with RSA accumulators.

**Milestone 4: Compare and Contrast and Write Dissertation (ISEC901 - 3)**

Scheduled Completion: [2026-2027]

Deliverable:

* Comparative analysis between the original Aleph protocol and the modified version with RSA accumulators.
* Dissertation document detailing the research methodology, findings, conclusions, and implications.
* Presentation of dissertation findings and defense.

**References**

Abraham, I., Jovanovic, P., Maller, M., Meiklejohn, S., Stern, G., & Tomescu, A. (2021). Reaching consensus for asynchronous distributed key generation. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing* (pp. 363–373).

Baird, L., & Luykx, A. (2020). The Hashgraph Protocol: Efficient Asynchronous BFT for High-Throughput Distributed Ledgers. *International Conference on Omni-layer Intelligent Systems* (pp. 1-7).<https://doi.org/10.1109/COINS49042.2020.9191430>

Boldyreva, A. (2002). Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *Public key cryptography–PKC 2003* (pp. 31–46). Springer.

Das, S., Xiang, Z., & Ren, L. (2020). Asynchronous data dissemination and its applications. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*.

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.ezproxylocal.library.nova.edu/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>

Fischer, M. J., Lynch, N. A., & Paterson, M. S. (1982). Impossibility of distributed consensus with one faulty process (Technical Report No. MIT/LCS/TR-728). Massachusetts Institute of Technology, Laboratory for Computer Science, Cambridge.

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)* (p. 1187–1201). Association for Computing Machinery.<https://doi.org/10.1145/3548606.3559379>

Gennaro, R., Jarecki, S., Krawczyk, H., & Rabin, T. (2003). Secure Applications of Pedersen’s Distributed Key Generation Protocol. In *Topics in Cryptology - CT-RSA 2003, The Cryptographers’ Track at the RSA Conference 2003, San Francisco, CA, USA, April 13-17, 2003, Proceedings* (pp. 373–390).<https://doi.org/10.1007/3-540-36563-X_26>

Gennaro, R., Jarecki, S., Krawczyk, H., & Rabin, T. (2007). Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. *Journal of Cryptology, 20*(1), 51–83.<https://doi.org/10.1007/s00145-006-0347-3>

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. doi:10.14722/ndss.2022.24385

Hood, K., Oglio, J., Nesterenko, M., & Sharma, G. (2021). Partitionable Asynchronous Cryptocurrency Blockchain. *IEEE International Conference on Blockchain and Cryptocurrency* (pp. 1-9).<https://doi.org/10.1109/ICBC51069.2021.9461080>

Hussein, K. M., & Al-Gailani, M. F. (2022). An Efficient Bandwidth Based on the Cryptographic Technique of the RSA Accumulator in Block Chain Networks. In 2022 Fifth College of Science International Conference of Recent Trends in Information Technology (CSCTIT) (pp. 164-168). Baghdad, Iraq. doi:10.1109/CSCTIT56299.2022.10145614

Kogias, E. K., Malkhi, D., & Spiegelman, A. (2020). Asynchronous Distributed Key Generation for Computationally-Secure Randomness, Consensus, and Threshold Signatures. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)* (pp. 1751–1767). Association for Computing Machinery.<https://doi.org/10.1145/3372297.3423364>

Knudsen, H., Li, J., Notland, J., Haro, P., & Ræder, T. (2021). High-Performance Asynchronous Byzantine Fault Tolerance Consensus Protocol. *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.

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

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

Ramakrishna Kotla, Lorenzo Alvisi, Michael Dahlin, Allen Clement, and Edmund L. Wong. (2009). Zyzzyva: Speculative Byzantine fault tolerance. *ACM Transactions on Computer Systems (TOCS), 27*(4), 7:1–7:39.<https://doi.org/10.1145/1658357.1658358>

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. doi:10.1109/COMSNETS51098.2021.9352892.

Silva, P., Matos, M., & Barreto, J. (2023). NimbleChain: Speeding up Cryptocurrencies in General-purpose Permissionless Blockchains. *Distributed Ledger Technology, 2*(1), Article 8.<https://doi-org.ezproxylocal.library.nova.edu/10.1145/3573895>

Woznica, A., & Kedziora, M. (2022). Performance and scalability evaluation of a permissioned Blockchain based on the 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>