Introduction:  
 The exploratory research for the Fall 2023 ISEC 885 course aims to develop a problem direction for a doctoral research idea concept paper, leading to an idea paper, with the ultimate objective of contributing to the field of Asynchronous Consensus**.**

Asynchronous Consensus refers to the ability of a distributed system to reach an agreement without any timing assumptions even in the events of faults, message delays, and unpredictable delivery times. Asynchronous Consensus ensures that systems can reach an agreement even when there is no bound on the time it takes for messages to be delivered or for nodes to respond. Not knowing the bound on the time in an asynchronous setting directly contrasts Synchronous Consensus, which does know its bound on time, along with Partially-Synchronous Consensus, which knows there is a bound on time, but doesn't know what that bound is (Miller et al., 2016 "Many Forms of Timing Assumptions," p. 3).

**Incorrect APA citation format; (Miller et al., 2016) is correct. Page numbers are for quotations. There is never a need for quotations.**

**The idea is to highlight the challenge of asynchronous vs synchronous.**

A problem with synchronous or partially synchronous protocols are the failures or performance degradation when network conditions are unpredictable, such as on the open internet. This is directly supported in the works of the Honey Badger Byzantine Fault Tolerant (HBFT) protocol whose research highlights the ability to bring down a well researched partially-synchronous protocol called Practical Byzantine Fault Tolerant (PBFT). This is done by the means of an adversarial network scheduler abusing the leader selection and halting progress by making the intermittent synchronicity connections too small for the network to catch up (Miller et al., 2016 "When Weak Synchrony Fails," p. 4).

**The concept of showing a fault in the synchronous protocol is fine, but where are you going with this to support a direction.**

**Incorret formatt**

In contrast, the advantage of asynchronous consensus and by nature the HBFT, is that during these smaller intermittent windows of synchronicity, the HBFT still makes progress. Highlighting the benefit that the HBFT can synchronize relatively instantaneously when the network connects again, in contrast to synchronous protocols which might not have a long enough synchronization window to catch up. It is important to note that the HBFT is significant to the realm of Asynchronous Consensus as it touts itself as

**Again, I get where you are going but this is too simplistic to provide support. Again, if you are justifying the early preamble to your concept, then just summarize the value of asynchronous protocols and their practical value in current communication.**

“the first practical asynchronous Byzantine Fault Tolerant (BFT) protocol, which guarantees liveness without making any timing assumptions.” (Miller et al., 2016 "Abstract," p. 1)

**Do not use quotations. You can express this in your thoughts and provide a standard citation.**

This is relevant because of how the HBFT’s Asynchronous Consensus relates to the Fischer, Lynch, Paterson (FLP) theorem. The FLP theorem argues you can't make an agreement in an asynchronous environment when crash failure are allowed, unless you augment the basic model in some way, such by adding randomization (Fischer et al., 1985). It is crucial to note that HBFT argues to practically get around the FLP theorem impossibility by using cryptographic techniques and assumptions to perform randomization. This is supported in future asynchronous consensus research such as in the newer asynchronous consensus protocol called Beat, who states “However, state machine replication cannot be achieved in asynchronous environments (Fischer et al., 1985), unless it uses randomization to circumvent this impossibility result. HoneyBadgerBFT and Beat fall into this category.” (Duan et al., 2018 “Related Work” p. 3 ) From this statement we can see research that supports the FLP theorem impossibility must be circumvented through randomization and that other research has supported HBFT’s claims of being one of the first practical asynchronous BFT protocols to do so. HBFT has historically been chosen as the bench mark in the field of Asynchronous Consensus through newer research such as Beat, Dumbo, Asynchronous Byzantine Fault Tolerance (ABFT), and the AlephBFT that will later be discussed (Knudsen et al., 2021,Duan et al., 2018, and Guo, Lu, Tang, Xu, & Zhang, 2020, Gągol et al., 2019).

**This must be reworked to show how HBFT and other asynchronous consensus protocols provide a valid protocol in this environment.**

Permission vs permission-less

Problem:   
 The current problem being that most modern research in the field of Asynchronous Consensus is implemented with a trusted dealer.

**This is not a valid problem statement.**

While having a trusted dealer is advantageous for permissioned blockchains they are not ideal for permission-less blockchains where it is assumed that all nodes on the network can be faulty as in the case of the open internet. The advantage of having a permissioned blockchain is that they typically make gains in security and scalability while giving up decentralization. On the other hand, permission-less blockchains typically have a higher decentralization factor with weaker scalability and security (Woznica & Kedziora, 2022 “Blockchain fundamentals” P.1)

**This is all over the place. You must make a case for why current communication models are in the realm of faulty nodes.**

The problem of the lack of decentralization in permissioned blockchains is directly argued the Aleph’s research stating that “in blockchain systems, no trusted entities can be assumed to exist, and thus a trusted setup is tricky if not impossible to achieve in real-world applications.” (Gągol et al., 2019 “Our Contribution” P. 4)

***Again, no need for the quotation. What is the problem this is moving towards supporting?***

Previous Asynchronous Consensus protocols admit to these fallacies, such as in the HBFT whose research admits to using a trusted dealer and solidifies that the known alternatives of using Distributed Key Generation (DKG) are synchronous in nature which is reinforced in their exert “we assume that nodes may interact with a trusted dealer during an initial protocol specific setup phase, which we will use to establish public keys and secret shares. Note that in a real deployment, if an actual trusted party is unavailable, then a distributed key generation protocol could be used instead (Boldyreva, 2002). All the distributed key generation protocols we know of rely on timing assumptions” (Miller et al., 2016 “Problem Definition: Atomic Broadcast” P. 5) The reliance on timing assumptions is what makes the previous works of DKG synchronous. The previous works of DKG being used in a synchronous manner is also supported in Aleph's research that states “There has been a lot of prior work on DKG (Gennaro et al., 2003 & Gennaro et al., 2007 & Kate et al., 2012 & Kotla et al., 2009), however none of the so far proposed protocols has been designed to run under full asynchrony.” (Gągol et al., 2019 “Verifiable Secret Sharing” P. 9) Relying on timing assumptions means that HBFT has a problem of being designed for asynchronous permissioned blockchains or giving up its asynchronous status to become permission-less and is once again supported in the works of Aleph who states “an unpleasant drawback of HBFT, especially in the context of trustless applications, is that it requires a trusted dealer to initialize.” (Gągol et al., 2019 “Introduction” P. 2) Using DKG in a manner that is advantageous for permissioned blockchains is also used in other Asynchronous Consensus research such as in HBFT’s predecessor the Beat protocol suite that states “Beat0 and all subsequent Beat instances allow efficient distributed key generation (Gennaro et al., 2007 & Kate et al., 2012), which should be run during setup.” (Duan et al., 2018 “Distributed Key Generation.” p. 7) Another example of using a trusted dealer being an issue in Asynchronous Consensus can be seen in the works of ABFT who admits that “One potential limitation of ABFT is that we do not implement the full threshold ECDSA scheme. The precomputing of the signing material is done using a trusted dealer. Related to this, we do not evaluate the cost of running the precomputing protocol for the threshold ECDSA scheme in the context of ABFT.” (Knudsen et al., 2021 “Limitations of ABFT” P. 7) From the works of ABFT we can see that they admit to using a trusted dealer making the protocol less suited for permission-less systems. Some more recent advancements of the trusted dealer issue can be seen in Aleph’s research who claims to be the first Asynchronous Consensus protocol without a trusted dealer (Gągol et al., 2019 “Abstract” P.1) and is done by creating a DKG via a proprietary trustless protocol “that performs the setup instead of a trusted dealer.” (Gągol et al., 2019 “Randomness Beacon with Trustless Setup” P.12) Aleph's research demonstrates two versions of the protocol. The trusted dealer version and the dealer-less version. The problem with the dealer-less version is that it is less cryptographically secure then its trusted dealer counterpart while arguing gains in decentralization for permission-less blockchain systems (Gągol et al., 2019 “Randomness Beacon with Trustless Setup” P.12). Aleph’s advancements in permission-less blockchain systems via asynchronous consensus without a trusted dealer has been solidified in the research of Dumbo whose “Related Work” states that “Very recently, Aleph and DAG-rider proposed to use direct acyclic graph for consensus besides sequential ACS, which provides a theoretical alternative for implementing asynchronous atomic broadcast besides sequentially executing ACS.” (Gao et al., 2022) and that “One interesting technique of Aleph is to remove the trusted dealer assumption.” (Gao et al., 2022) From the above research we can conclude that current problem with Asynchronous Consensus protocols is that a majority of them rely on a trusted dealer making them less advantageous for permission-less blockchain systems and more suited for permissioned blockchains. Noting, that the current permission-less asynchronous research provided by Aleph admits to still being less secure then its trusted dealer counterpart. Providing, a need for further research in the field of Asynchronous Consensus in permission-less systems without a trusted dealer.

***Not sure where this is going.***

***As previously mentioned, you must not use quotations. As this is presented it does not establish a clear and clean flow that builds a case for supporting a problem and need.***

***You are listing excerpts and quotations but not focusing on a problem and need. Other than highlighting that protocols use a trusted dealer, this is just a survey of the literature to provide a feature of current protocols such as a trusted dealer.***

***Aleph does not use a trusted dealer but is less secure. So it is not clear what the problem is as you have presented it other than to extend the Aleph work to address the need for non-trusted dealers in current permission-less blockchain environments and achieve a definable (from the literature ) level of security.***

Goal:  
 The goal of this exploratory research, conducted during the Winter 2024 ISEC 885 course, is to formulate a research idea paper focused on advancing the field of Asynchronous Consensus by developing a replicatable framework for deploying the Aleph protocol that is without a trusted dealer and is well suited for permission-less blockchains. This framework will establish a foundational benchmark for future studies, enabling researchers to compare and to build upon the original Aleph work effectively. While the existing Aleph paper offers comprehensive insights into the theoretical underpinnings of its consensus protocols, it lacks a practical framework that can be readily replicated for further investigation. Thus, the objective of this research is to furnish the Asynchronous Consensus community with a standardized test environment for Aleph, facilitating ongoing research and development endeavors.

**This must be preceded by a problem. Just providing an environment that supports a protocol is not a dissertation problem/goal pair.**

**You must establish a problem with existing frameworks to then focus on building one. There is a mismatch in the basic concept of the previous section and this one to solidify a concept paper.**

**There must be more than just implementing an environment that supports a baseline of current protocol work such as Aleph.**

References

Adam Gągol, Damian Leśniak, Damian Straszak, and Michał Świętek. 2019. Aleph:

Efficient Atomic Broadcast in Asynchronous Networks with Byzantine Nodes.

In Proceedings of the 1st ACM Conference on Advances in Financial Technologies.

214–228.

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.  
  
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.

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  
  
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  
  
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  
  
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  
  
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  
  
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.