Dear Dr. Lakmal Rupasinghe

I am writing to update you on the progress of my research project entitled Vulnerabilities of smart contracts using case. Please find below the details of my progress, including my contributions, references, and other important elements.

Project Progress:

A. Overview: We decide to find solutions for the current vulnerabilities of smart contracts. The project on vulnerabilities of smart contracts aims to identify and address security risks in blockchain-based applications. Smart contracts are self-executing programs that run on a blockchain, and they have the potential to revolutionize various industries by automating tasks and eliminating intermediaries. However, they are vulnerable to various security issues, such as coding errors, design flaws, and malicious attacks, which can result in the loss of funds and other damages. The project involves analyzing smart contract vulnerabilities, developing tools and techniques for detecting and preventing them, and educating developers and users about best practices for secure smart contract development and deployment.

B. Accomplishments: We create a topic assessment, and we are waiting to evaluate our topic assessment. I gather much information about my sub-component (race condition) by referring to the research papers and analyzing those data.

C. Challenges: There are some conflicts with other sub-components of our project.

D. Next Steps: Find a way to apply race condition solutions to the smart contracts by creating them for suitable as run-in smart contracts.

D. Contributions: I read “Aggregating Atomic Clocks for Timestamping” and “Detecting Race Conditions in One-Sided Communication of MPI Programs”. I found some race condition detection methods in the “Detecting Race Conditions in One-Sided Communication of MPI Programs” research paper. Such as the IMB benchmark program. But these methods use only for one-sided communication on MPI (Message passing interface) programs. In this detecting method, first divide windows using window sizes and then process get buy each window. This process is called MPI\_WIN\_FENCE. For example, if windows are already dedicated to writing operations and other operations also try to allocate that window using this IMB Benchmark program can detect it. But this is a preventive technic in this research paper. There is a prevention technic in this research paper called MPI\_WIN\_LOCK. This technic prevents access processes enter until complete the current process. Another solution is MPI\_WIN\_Flush technic. In this technic preparation1 process I determinist order and prevent this race condition situation. Another technic is the dynamic load balancing technic this technic distributes workload among processes and does quicker tasks and prevents race conditions. And next, I should find how to engage this with smart contracts and blockchains. So, I refer “Aggregating Atomic Clocks for Timestamping “research paper this research paper is mainly about how to manage the latency of the processes. From this research paper, I could find one prevention method using hash value calculation of the blocks and preparing an order of the processes by completing the order.

E. Collaborations:

References: