Abstract

In quorum selection, a large group of parties wants to select a small subset of its members, so that the subset can perform computations securely that would be too expensive for the entire group to perform. However, this problem is difficult since not all the parties are trustworthy. We assume there is an adversary who can control up to a third fraction of the parties. The goal is to select the members of the quorum at random, so that selection of the quorum must be sufficiently random, so that malicious, cooperating adversaries are unable to interfere with computations performed in the quorum. However, little work has been done in implementing and improving the practical performance of these protocols.

Furthermore, if the quorum is performing long-running calculations, then parties should be able to enter and leave and system, a phenomenon known as churn. Churn adds complexity because the adversary is able to replace honest parties with malicious parties during the protocol. Some work has been done in addressing churn, however, the new protocols have yet to be benchmarked (Commensal Cuckoo).
For many distributed algorithms, the computation cost and communication cost protocols scale linearly or superlinearly with the number of parties. These protocols would be infeasible for sufficiently large groups of parties (some examples of important protocols that don’t scale well). Therefore, the parties need a mechanism to select a subset of parties. On the other hand, protocols must be secure against the malicious adversary that controls a number of the parties and can deviate arbitrarily from the protocol. We assume that the proportion of malicious parties in the whole group is less than 1/3. However, in order for the quorum to be sufficiently small to efficiently do expensive computations, the number of malicious parties in the group exceeds the threshold for computations among the quorum. Therefore, if malicious parties are able to influence the quorum selection process, they would be able to interfere with computations in the quorum.

Churn is another important consideration. The protocol must be secure when the adversary is able to choose honest parties to leave and malicious to enter, with the restriction that the total number of malicious parties is always lower than the threshold. Since each quorum is small compared to the total number of parties, Specifically, with quorums, the adversary could remove honest parties from a quorum in an attempt to increase the proportion of malicious parties to an insecure level. While churn adds complexity, large, long running systems, which justify an expensive protocol to set up quorums must be able to address parties entering and leaving in order to be practical.

Project

My research will consist of working with Mahnush Movahedi, who is a post-doc. I will work with Mahnush and Professor Raykova to design, implement and refine a protocol that can be used to perform quorum selection. We will place heavy emphasis on practical performance
with a large number of parties. The first step will be to design and implement a simple framework that can accurately measure the performance of the protocol in terms of time, computation cost, and bandwidth usage. Concurrently, we will design a protocol that meets the security requirements in a setting with malicious parties and churn. Second, I will implement the designed protocol in the framework and test its performance. Since these protocols have not yet been benchmarked with large numbers of parties, we expect that the design process will require multiple iterations. In each iteration, we will modify a portion of the protocol while preserving or improving its security and update the test implementation to match the new protocol.

References


Deliverables

- Protocol and security proof for quorum selection - 5 weeks
- Framework for multi-party computation - 2 weeks
- Implementation of protocol in the framework - 2 weeks
- Data on the performance and scalability of the new protocol - 1 week