Computing without a Leader: Building Blocks for Internet-Scale, Robust Computing

Motivation and Problem: How do ant colonies, bee hives, and markets function even when there is no leader? A starting point for answering this question is the fundamental problem of agreement in distributed computing: *Byzantine agreement*. The Byzantine agreement problem is to devise a protocol so that n agents, each with a private input can agree on a single common output that is equal to some agent's input. A certain unknown subset of the agents suffer *Byzantine faults*: they can engage in any kind of deviations from the protocol, including false messages and collusion.

Byzantine agreement has found applications in many areas. Unfortunately, continued application is hampered by a stark barrier: there is no practical, scalable algorithm for Byzantine agreement. In particular, all current Byzantine agreement algorithms for the asynchronous communication model require all-to-all communication.

Intellectual Merit: The proposed research will directly address this barrier by designing scalable algorithms for Byzantine agreement and other related problems. Our goal will be to design algorithms that are scalable in the sense that each agent sends a number of bits that is $O(\sqrt{n}\log n)$, and total latency is $O(\log n)$; and robust in the sense that they can tolerate up to a constant fraction of faulty processors, where these faults may either be Byzantine faults, or *fail-stop faults*: the processors simply stop participating in the protocol. In addition to Byzantine agreement, we will design scalable and robust algorithms for the following related problems:

- *Subcommittee Election:* All processors agree on one or more subcommittees of size $O(\log n)$, where the fraction of bad processors in each subcommittee is within ϵ of the fraction of bad processors in the network, for any positive ϵ.
- *MapReduce:* Enable the MapReduce software framework, even when there is no master.
- *Robust Multiparty Computation:* Each processor starts with a private input and there is a publicly known function F on n variables. The goal is for all users to learn the output of F at the point given by the private inputs. Note: Unlike in secure multiparty computation, it is not required to maintain the privacy of the inputs.

This project will build on new algorithmic and mathematical techniques we have developed in recent results that break through some conjectured bottlenecks for Byzantine agreement with synchronous communication. These new techniques include the use of 1) *averaging samplers:* a type of bipartite graph that ensures bad agents are "well-spread" in the network; 2) (s,t)*-random sources*: a new distributed data structure that outputs s bits, at least t of which are guaranteed to be random; 3) *iterated secret-sharing*: a scheme that ensures arrays of random bits initially held by a processor are split among an increasingly larger numbers of processors as the array becomes more important during the computation.

Broader Impact: The goal of this proposal is to develop building blocks for creating reliable, large-scale distributed systems. The long-term vision is to develop a technique, based on agreement, that is on par with techniques like cryptography and error-correcting codes by 1) being frequently used in practice and applicable across a wide range of applications; 2) having a clean interface between theory and practice that provides a) practitioners with a set of well-studied algorithms with clear, provably good solutions to fundamental problems; and b) theoreticians with a set of formally defined mathematical and algorithmic problems that will have direct practical importance. Success in this endeavor will likely have an impact across many areas including: peer-to-peer systems, database systems, control systems, grid computing, cloud computing and game theory.

Risk: This is a high-risk, high-payoff proposal. Byzantine agreement is a challenging problem which many strong researchers have worked on for almost as long as computer science has been an academic discipline, and as yet there are no truly scalable algorithms. We argue that the new mathematical and algorithmic techniques we have developed partially mitigate our risk of failure. Moreover, we argue that this risk is justified by the importance of the problem; if there is ever to be a science of trustworthy computing, we need to solve such challenging, fundamental problems.