Galois Proposed Project 2

Distributed Verifiable Elections

Problem Description and Project Purpose

Most verifiable elections systems are based upon a static client-server model. Trusted parties run a set of election servers and voters—either using their own computers (for early voting or voting from home) or using state-provided kiosks (for traditional supervised voting in polling places)—run or use client voting terminals.

A few variants of verifiable election algorithms tackle the problem using more novel distributed algorithms [1]. These algorithms either involve an arbitrarily large set of processes whose provenance is unknown (think "BitTorrent [2] for elections" or "elections over Tor [3]") or use the BitCoin blockchain [4]. None of these novel algorithms have been developed, deployed, and exercised to determine their in-the-field capability and utility. The focus of this project is on doing just that.

Actions and Deliverables

- Learn about a small number of novel distributed verifiable election schemes.
- Design and develop a demonstrator version of one scheme. If the scheme is a classic distributed algorithm, the development should either be done in Haskell [5] on HaLVM [6] (if the developer students have appropriate expertise) or in any programming language and deployed on a cloud platform like AWS [7] or Heroku [8].
- Measure the behavior of the deployed system under various deployment configurations and loads (e.g., 1, 10, 100, 1000 instances with 1, 10, 100, 1000 client interactions per second).

Resources and Support

Galois will provide remote deep technical R&D assistance on matters relevant to our expertise. Galois can pay for compute time on cloud services if necessary.

[1] http://en.wikipedia.org/wiki/Distributed_algorithm
[2] http://en.wikipedia.org/wiki/BitTorrent
[3] https://www.torproject.org/
[4] http://en.wikipedia.org/wiki/Bitcoin#The_block_chain
[5] https://www.haskell.org/
[6] https://galois.com/project/halvm/
[7] http://aws.amazon.com/
[8] https://www.heroku.com/

Expertise Necessary

At least one student on a team executing on this project must have some distributed programming experience.