A Decentralized and Distributed E-voting Scheme Based on Cryptographic Shuffles

Decentralized and Distributed Systems Laboratory

Andy Caforio
Responsible: Prof. Bryan Ford
Supervision: Linus Gasser, Philipp Jovanovic
Way back when...
Helios

- Started in 2008
- First web-based, verifiable e-voting scheme
- Leverages cryptographic shuffles
Helios - Features

- Auditable elections
  - Encryption proof
  - Shuffle proof (Sako-Kilian)
  - Decryption proof
- User authentication
- Front- and back-end implementation
Helios - Verifiability

- Users can verify that their vote was counted
- Shuffle weeds out malicious servers
- Honest servers will perform decryption
Helios - Protocol

1. Cast
2. Publish
3. Shuffle
4. Audit
5. Decrypt
6. Tally
Helios - Disclaimer

- Helios does not enforce anonymity
- Voters may be subject to coercion
Helios - Downsides

- Centralized
- Conventional database storage
- Very slow shuffles
Helios - Improvements

- Centralized Cothority
- Conventional database storage Skipchains
- Very slow shuffles Neff
Back to the future

Cothority

Protocols

DKG  Neff  Decryption

Skipchains

Master → Link → Link

Election → Ballot → Ballot → Shuffle → Decrypt
Protocols - DKG

- Distributed Key Generation
- Create public/private key pair
- Split private key
- Part of kyber library
Protocols - Neff Shuffle

- Novel verifiable shuffle concept by Andrew Neff
- Orders of magnitude faster than Sako-Kilian scheme
Protocols - Decryption

- After election termination and audit
- Reconstruct plaintext ballots with shared secret keys
- Cannot be done by a single node
Storage

- **Master**
  - System configurations
  - List of admins, roster etc.

- **Link**
  - Reference to election skipchain

- **Election**
  - Settings
  - DKG public key, list of voters etc.

- **Ballot**
  - Casted vote (one per block)

- **Shuffle**
  - Permuted and re-encrypted ballots

- **Decrypt**
  - Ballot plaintexts
Practical

- Go implementation
- Built on top of cothority and kyber
- Protobuf API
Benchmarks - Helios

- Shuffle of 500 ballots ~130s
- 2.2 GHz dual core machine

[Ben Adida. Helios: Web-based open-audit voting]
Benchmarks

- Shuffle
- 1.4 GHz dual core
- Real world context?
Overview

● Distributed e-voting scheme
● Improves on Helios
  ○ Distributed
  ○ Faster
● Built on top of DEDIS infrastructure
Gory details

- **Cryptographic background**
  - Framework (elliptic curve etc.)
  - Shuffles
  - Verifiability

- **Protocols**
  - Networking

- **Usage**
  - Authentication
  - Front-end
References

- Repository: https://github.com/dedis/student_17_evoting