LYU1803:

Opensource E-voting System for 8 million mobile devices

ESTR4998 Graduation Thesis Presentation

Maxwell Chan presents

supervised by **Prof. Michael Lyu**

Introduction

Motivation

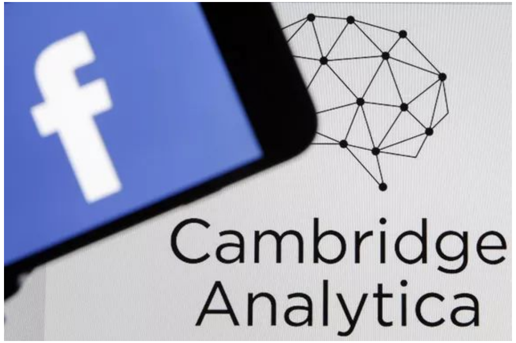
Paper-based voting

- Time and resources
- Disencourage voter
- Harm democracy

Motivation

Mistrust

- Public, Government, Computer
- Government controls computer → Public cannot monitor
- Network security / personal data leak incidents

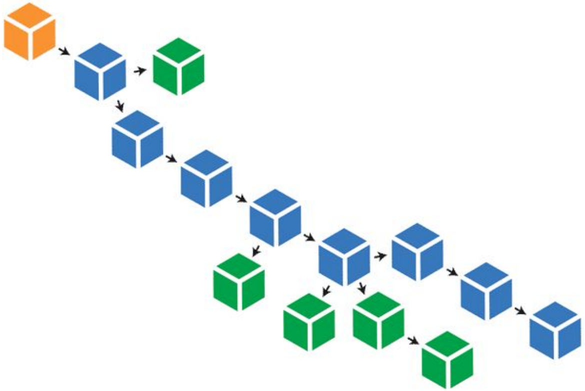


Cathay Pacific Data Breach Exposes 9.4 Million Passengers

Motivation

Blockchain

- Popular nowadays
- Reliable & trusted data
- Transparency, auditability, decentralization, ...



⇒ Voting + Blockchain

Background

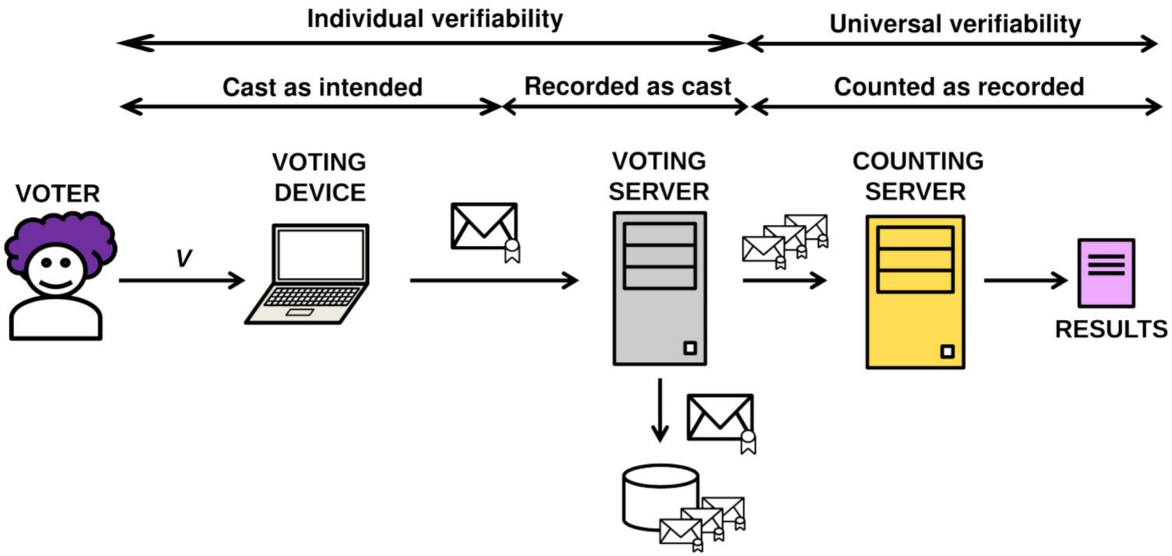
1. E-voting consideration

2. Blockchain

E-voting consideration

End-to-end verifiability

- Promote overall integrity



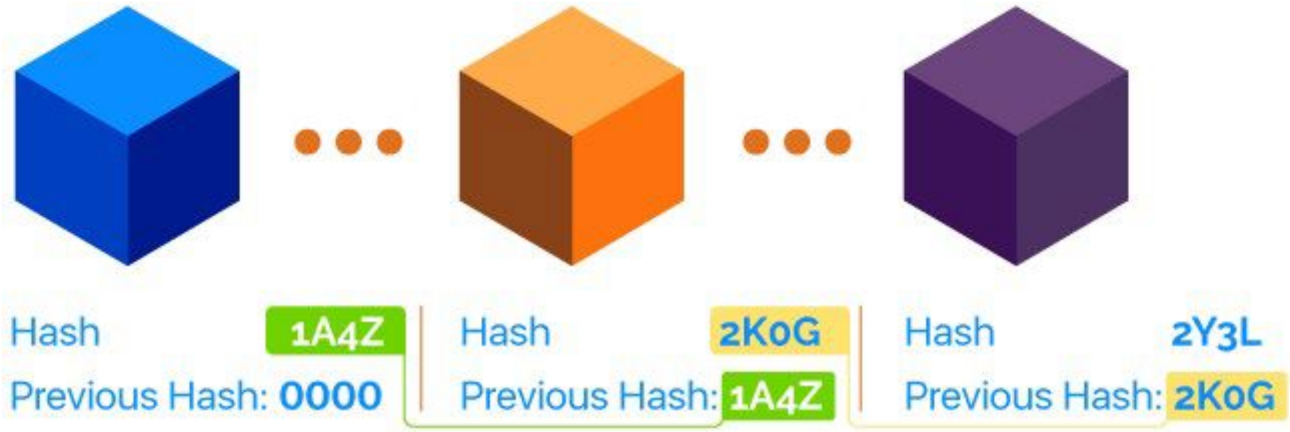
E-voting consideration

Authentication

- Only eligible voter can vote
- Ballot should be anonymous

Blockchain

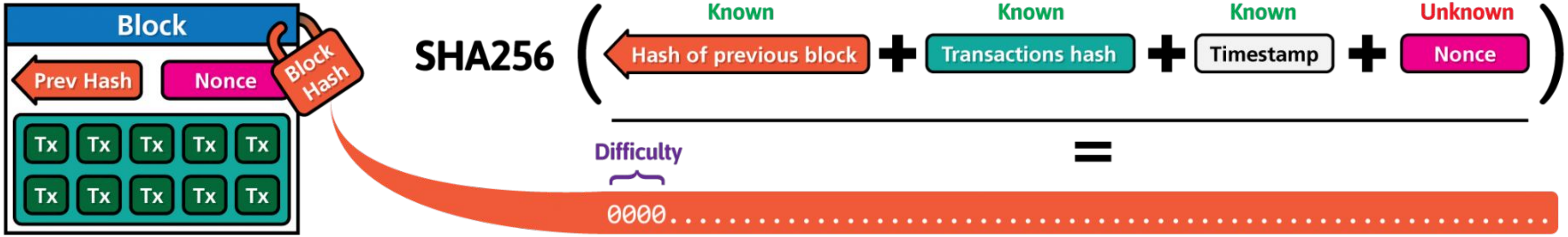
- A way to store data
- Non-modifiability
- Distributed & decentralized → need consensus



Blockchain

Permissionless blockchain

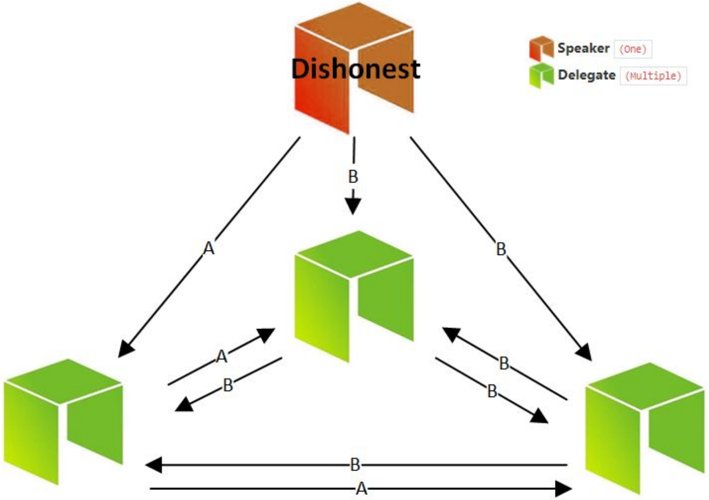
- Proof-of-work



Blockchain

Permissioned blockchain

- Byzantine Fault Tolerance



Objective

Goal

- E-voting application
- Satisfy e-voting consideration
- Use blockchain technology

⇒ Transparent & reliable e-voting for public

1st Term

- Explore and study
- System design
- Basic implementaion

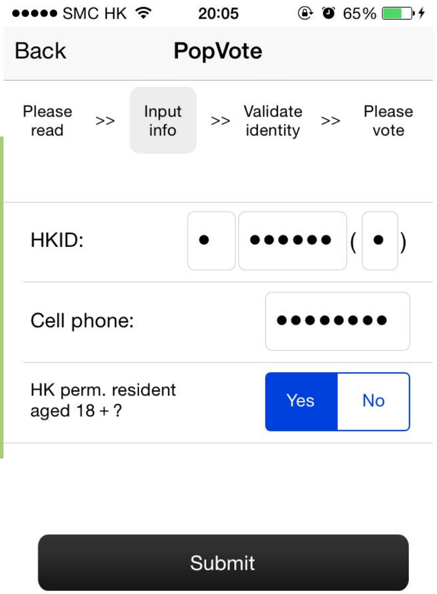
Related work

E-voting in Hong Kong

NO end-to-end verifiable system

Popvote

- Civil referendums
- Reports on security loopholes


●●●●● SMC HK 20:05 65%

Back PopVote

Please read >> Input info >> Validate identity >> Please vote

HKID: ● ●●●●●● (●)

Cell phone: ●●●●●●●●

HK perm. resident aged 18 + ? Yes No

Submit

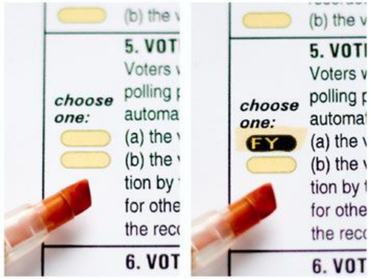
Related Work

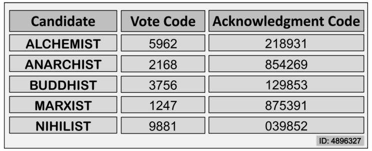
End-to-end verifiable voting system

Prêt à Voter, Scantegrity, Punchscan, Pretty Good Democracy, ...

Helios

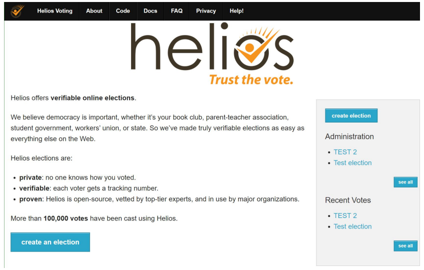
- Opensource + online implementation + remote voting
- Trustees: private keys
- Ballot fingerprint → ballot bulletin board
- Decrypt aggregation → Not single ballot





Candidate	Vote Code	Acknowledgment Code
ALCHEMIST	5962	218931
ANARCHIST	2168	854269
BUDDHIST	3756	129853
MARXIST	1247	875391
NIHILIST	9881	039852

ID: 4896327



Helios offers **verifiable online elections**.

We believe democracy is important, whether it's your book club, parent-teacher association, student government, workers' union, or state. So we've made truly verifiable elections as easy as everything else on the Web.

Helios elections are:

- **private**: no one knows how you voted.
- **verifiable**: each voter gets a tracking number.
- **proven**: Helios is open-source, vetted by top-tier experts, and in use by major organizations.

More than **100,000 votes** have been cast using Helios.

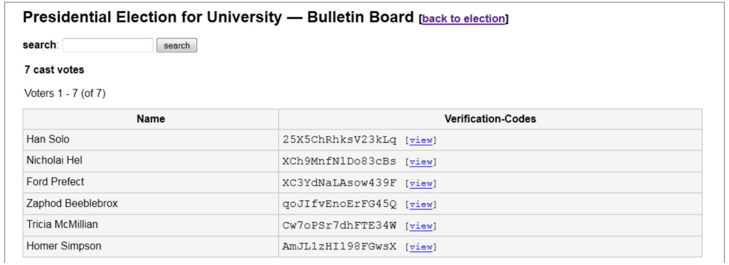
create an election

Administration

- TEST 2
- Test election

Recent Votes

- TEST 2
- Test election



Presidential Election for University — Bulletin Board [back to election]

search:

7 cast votes

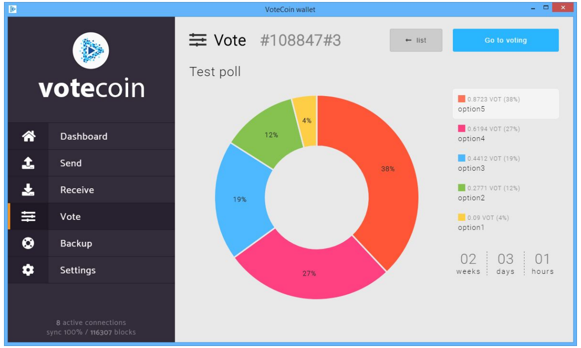
Voters 1 - 7 (of 7)

Name	Verification-Codes
Han Solo	25X5ChRhksV23kLq [view]
Nicholai Hel	XCh9MnfNlDo83cBs [view]
Ford Prefect	XC3YdNaLAsow439F [view]
Zaphod Beeblebrox	qoJIfvEnoErFG45Q [view]
Tricia McMillian	Cw7oPSr7dhFTE34W [view]
Homer Simpson	AmJL1zHI198FGwsX [view]

E-voting using blockchain

1 vote = 1 coin

- Intermediate result
- Provable intention



🟧 0.8723 VOT (38%) option5
🟥 0.6194 VOT (27%) option4
🟦 0.4412 VOT (19%) option3
🟩 0.2771 VOT (12%) option2
🟨 0.09 VOT (4%) option1

02 weeks 03 days 01 hours

Ballot as data

- Secure storage

Design

Overview

Helios - as reference

- Cryptography
- Limitation & Modification

Blockchain - as secure storage

- Type
- Protocol design

Cryptogrpahy

Homomorphic El Gamal encryption

Create election

p: a prime number

g: a primitive root of p

For each trustee:

private key: x_i, $0 < x_i < p - 1$

public key: $y_i = g\text{^}(x_i)\ \mathit{mod}\ p$

Election public key:

$$y = y_1 y_2 y_3 \ldots\ \mathit{mod}\ p$$

Public: $\{p, g, y\}$

Private: $\{x_1, x_2, x_3, \ldots\}$

Cryptogrpahy

Prepare ballot

For each option in each question:

Public: $\{p, g, y\}$

if voter choose this option, $i = 1$; else $i = 0$

$$m = g^i \bmod p$$

random number: r, $0 < r < p - 1$

$$c_1 = g^r \bmod p \qquad c_2 = y^r m \bmod p$$

Encrypted option: $\{c_1, c_2\}$

Cryptogrpahy

Compute result

For each option in each question:

Aggregation:

Encrypted option of voter a: $\{c_{1,a}, c_{2,a}\}$

$$c_1 = c_{1,1}c_{1,2}c_{1,3}\ldots \; mod \; p \qquad c_2 = c_{2,1}c_{2,2}c_{2,3}\ldots \; mod \; p$$

Decryption:

Public: $\{p, g, y\}$, Private: $\{x_1, x_2, x_3,\ldots\}$

$$g^m = c_2\,(c_1\text{^}(x_1)c_1\text{^}(x_2)c_1\text{^}(x_3)\ldots)^{-1} \; mod \; p$$

Result: $\boldsymbol{m}$ (discrete logrithm on g^m base g)

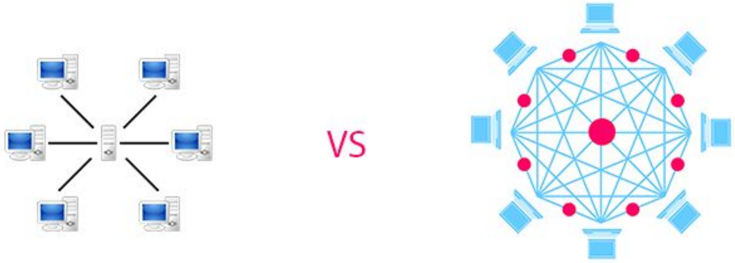
Limitation & Modification

Denial of service attack

- Single server / database
- Single point of failure

⇒ Blockchain

- Distributed
- Many copy
- Better trace



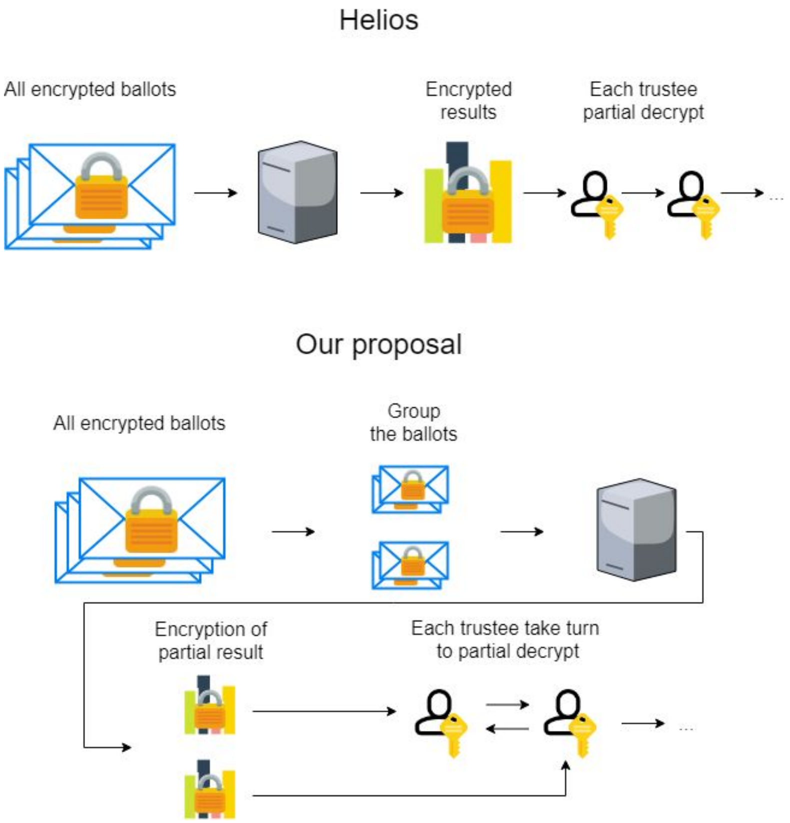
Limitation & Modification

Slow tally

- Aggregation
- Discrete logrithm

⇒ Allow decrypt in batch

- Won’t violate anonymity



Limitation & Modification

Coercion

- Voter prove to coercer
- Coercer sits next to voter
- Voter give out his credentials

Helios: allow re-voting

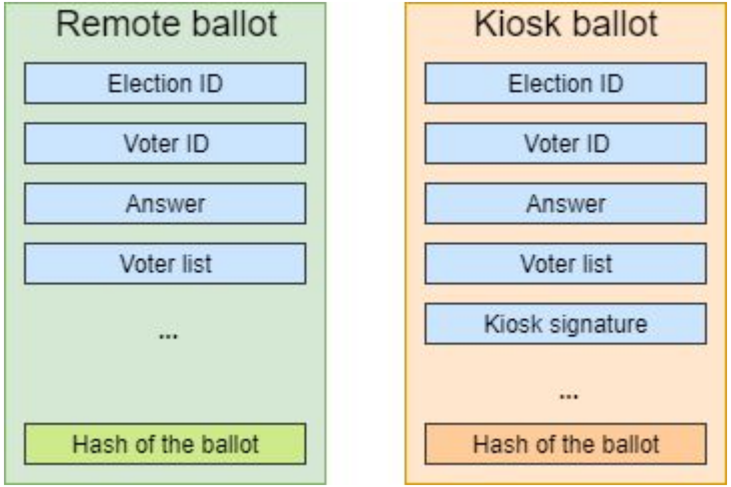
Limitation & Modification

Coercion

- Keep re-voting mechanism

⇒ Option for in-person voting

- Setup kiosk
- Higher priority
- Coercion risk ∝ Election scale



Remote ballot

- Election ID
- Voter ID
- Answer
- Voter list
- ...
- Hash of the ballot

Kiosk ballot

- Election ID
- Voter ID
- Answer
- Voter list
- Kiosk signature
- ...
- Hash of the ballot

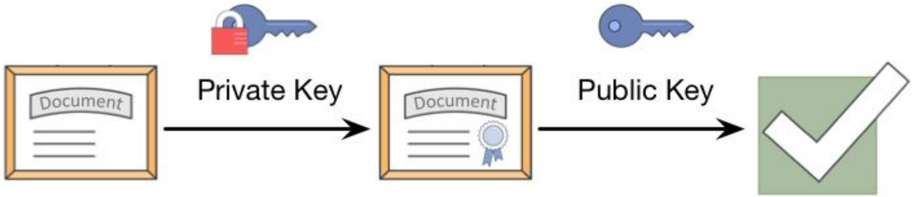
Limitation & Modification

Authentication

- Google / Facebook
- No public verification

⇒ Ballot signature

- RSA key pair for each voter
- Private key sign the hash



Limitation & Modification

Authentication

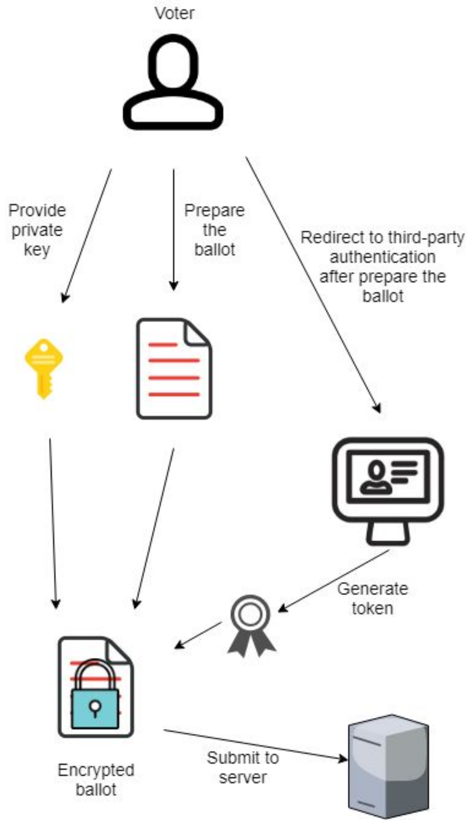
- Key owner = user?

⇒ Suggest further authentication

- Use valuable credential

⇒ API

- Generic for different election
- Third-party authentication



Limitation & Modification

Knowledge of who has voted

- Ballot bulletin board
- Obvious voter intention → problematic

⇒ Not guessable voter ID

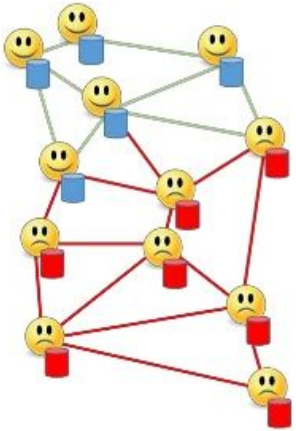
⇒ 'Abstention' option

⇒ Don't disclose voter ID

Type of Blockchain

Permissionless

- 51% attack
- Computationally intensive consensuses



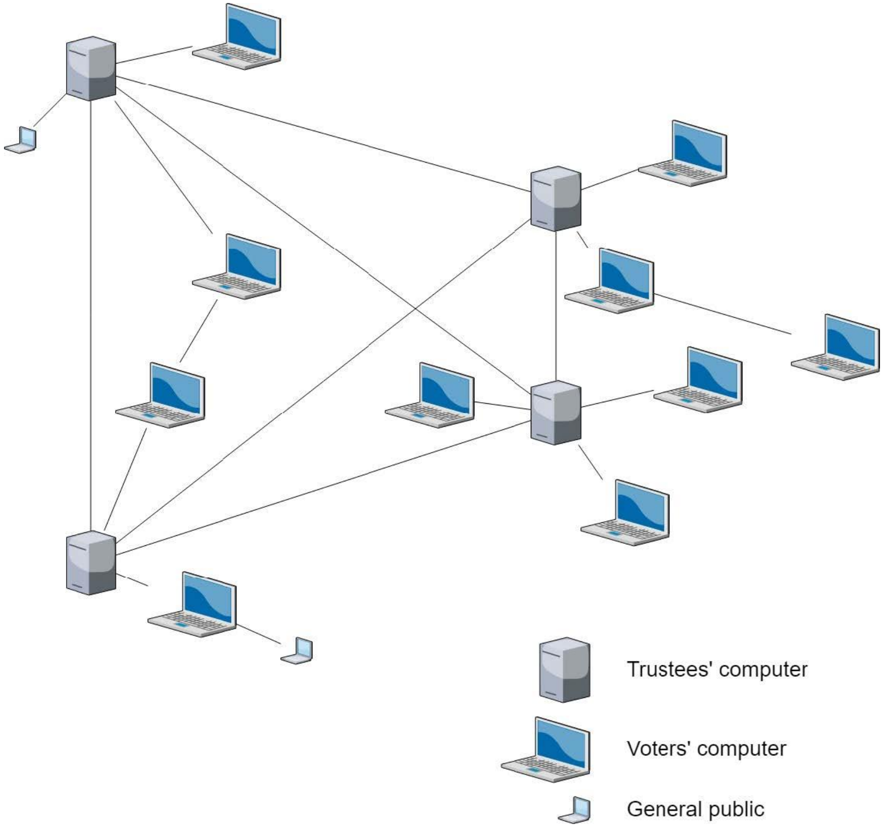
Permissioned

- Trust on trustee
- Allow private election

Roles and permission

Trustee: read + write

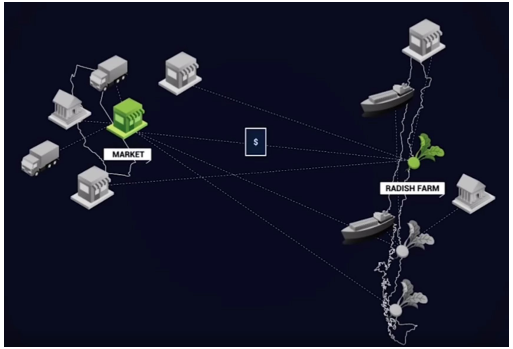
Voter / public: read



Design a blockchain protocol for voting

Opensource library

- Not many available
- ‘Hyperledger Fabric’
- Security loopholes

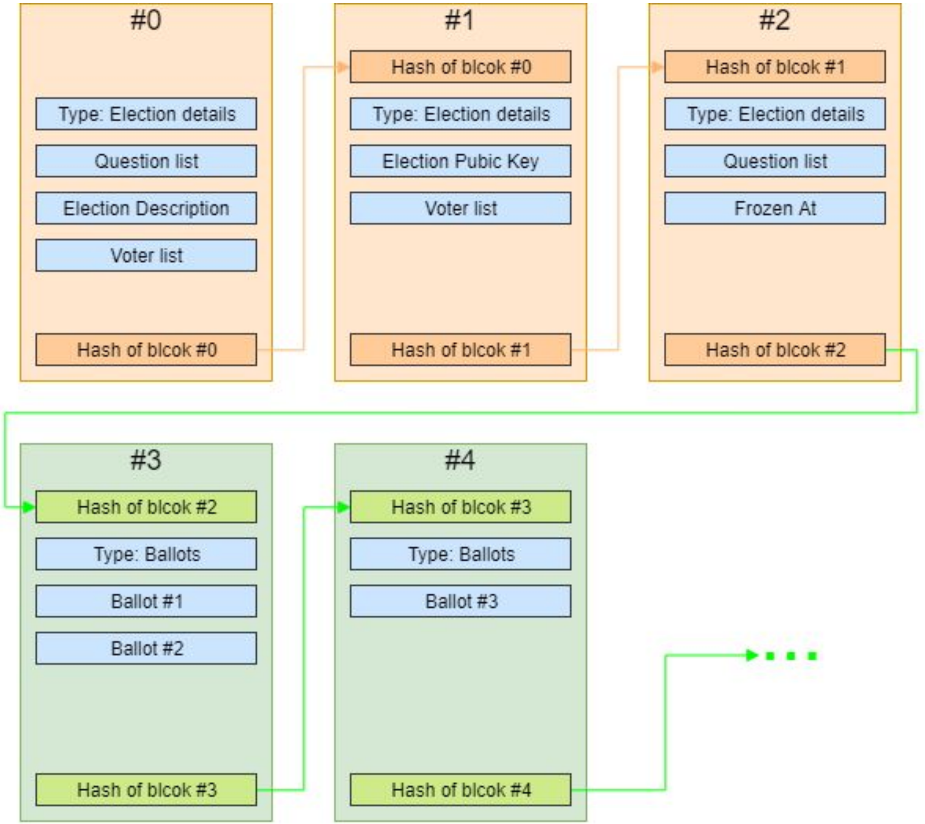


Define our protocol

- Lightweight
- Fit for voting
- New vulnerabilities → Opensource

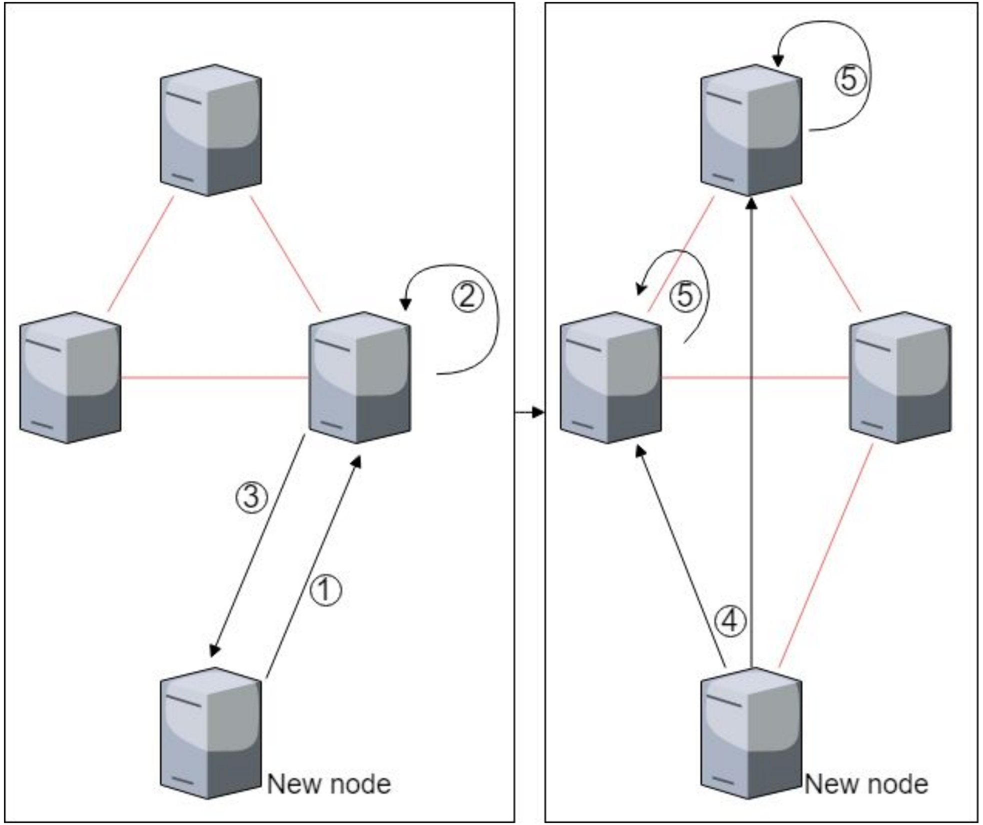
The Blocks

- 1 blockchain for 1 election
- ‘Election details’ & ‘Ballot’ blocks
- ‘Ballot’ block generated in a regular time interval



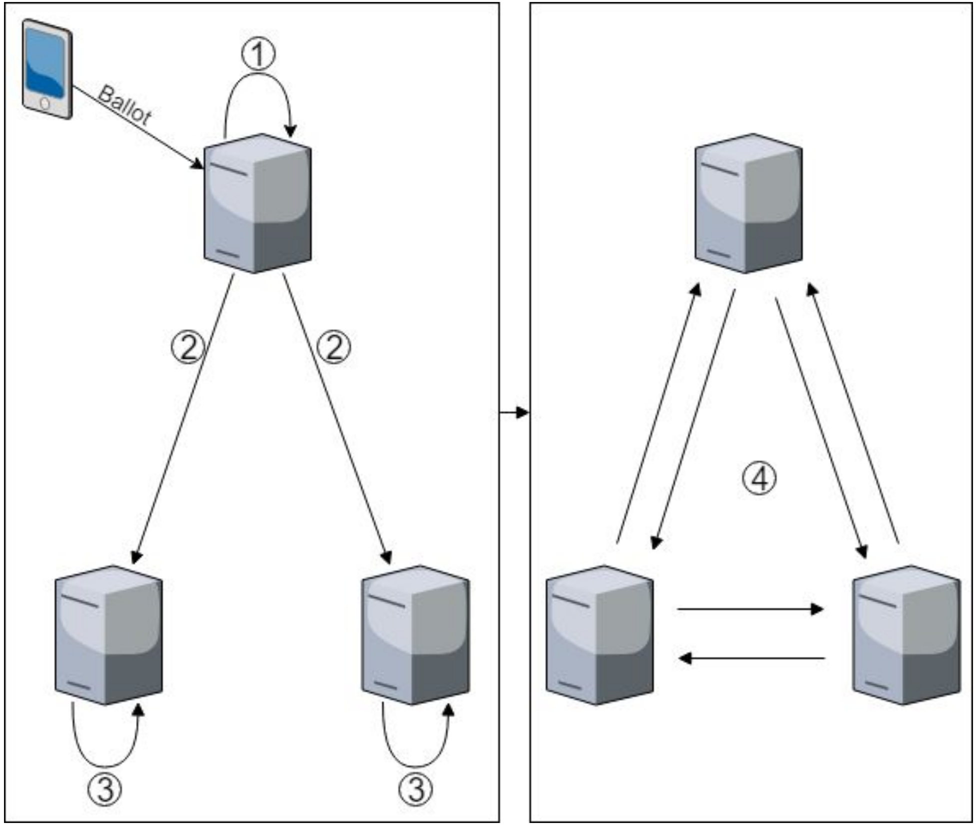
Handshake

- Every trustee’s node connect to each other
- Ping periodically



Ballot submission

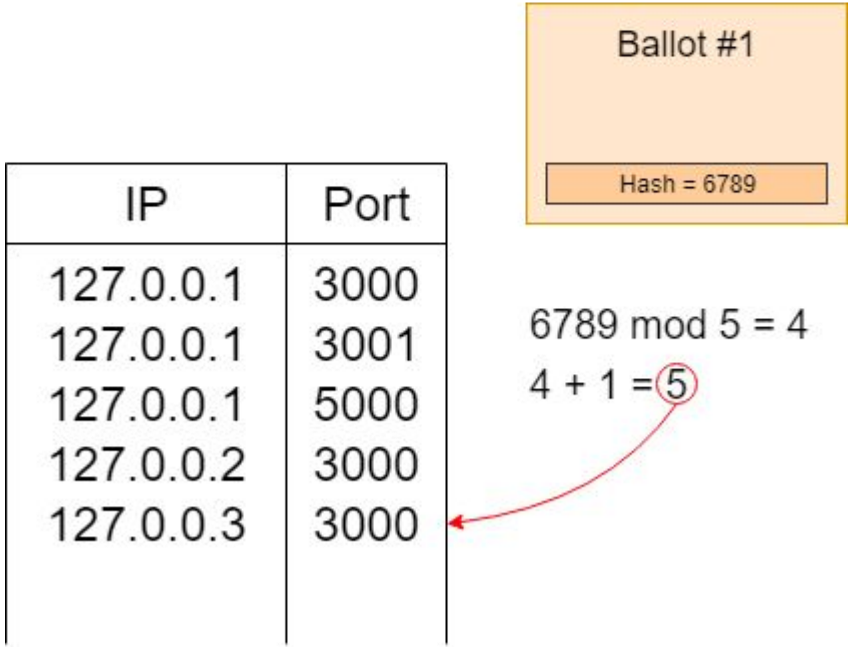
- >½ trustees sign → verified



Block generation

Node selection

- Use ‘last verified ballot’ with time buffer
- Nodes join/leave network → Result may be different



IP	Port
127.0.0.1	3000
127.0.0.1	3001
127.0.0.1	5000
127.0.0.2	3000
127.0.0.3	3000

Ballot #1

Hash = 6789

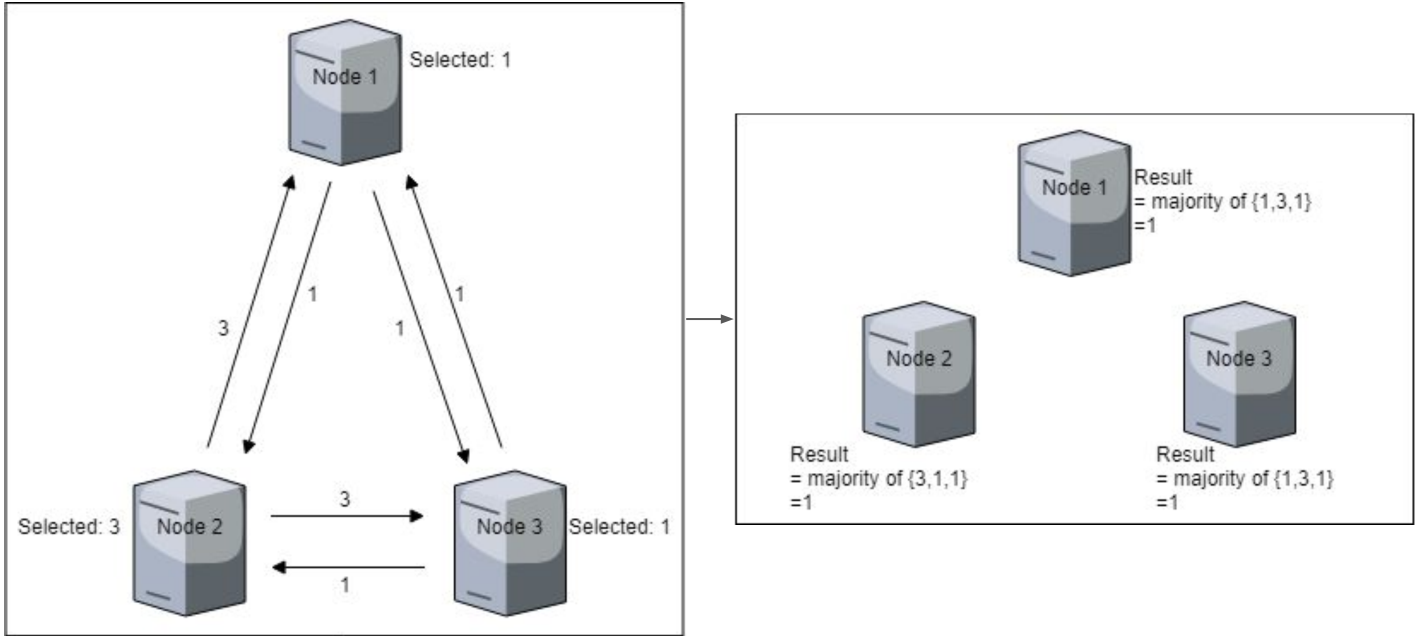
6789 mod 5 = 4

4 + 1 = 5

Block generation

Consensus

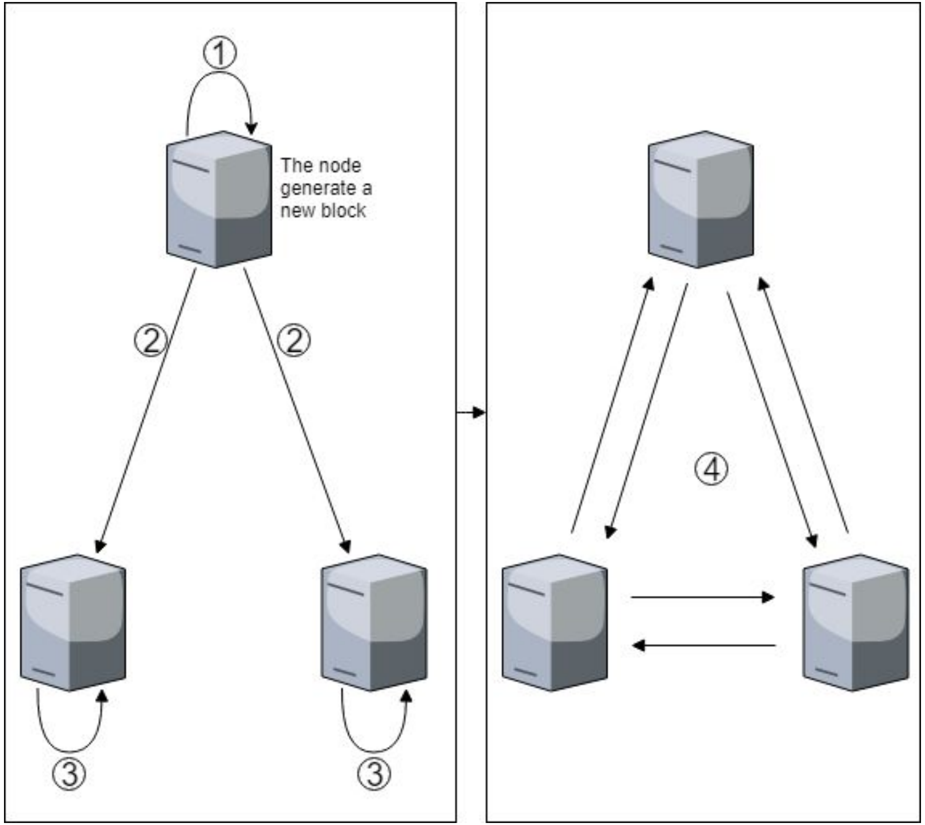
- Byzantine Fault Tolerance algorithm



Block generation

Block broadcasting

- >½ trustees sign → block verified → blockchain



Implementation

Overview

Client-side (voter / election organizer)

- Create election
- Vote
- Compute result
- Almost like Helios, except user-friendly interface

Server-side (trustee’s nodes)

- Connect to each other
- Broadcast ballot
- Generate & broadcast block
- Voting-related function

Demo

1. Connecting nodes
2. Create an election
3. Vote in the election
    - Ballot validation & broadcast
    - Block generation & broadcast
4. Compute result

Conclusion

Summary

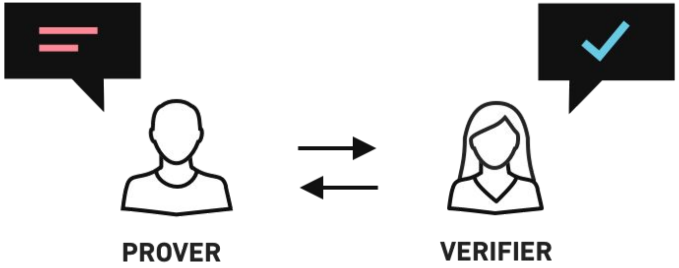
- Studied on end-to-end voting / blockchain voting
- Proposed modification to Helios & Designed blockchain protocols
- Basic implementation

Planned work

1. Zero-knowledge proof
2. Full blockchain verification
3. User interface
4. Apply proposed modification

Zero-knowledge proof

- Proving someone knowledge without learning other information
- Implemented in Helios



Zero-knowledge proof

Trustee knowledge on private key

- Unable to decrypt the election
- Fraud a public key → Decrypt all ballots himself

Trustee honest decryption

- Manipulate ciphertext → Modify election result

Voter honest encryption

- Encrypt invalid value → Affect the result

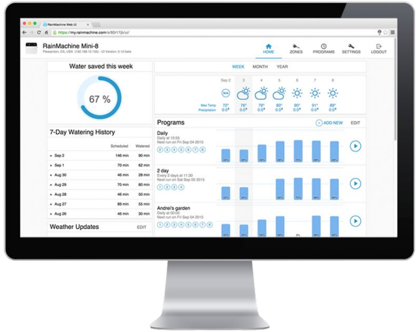
Full blockchain verification

- Ballots re-verification in new block
- Trustee’s signature verification
- Connection request validation
- ...

User interface

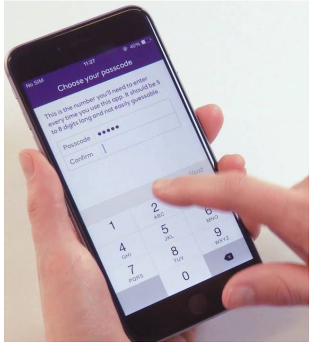
Web application

- Portable
- No installation
- Simpler → Work on other aspects



Mobile application

- Personal device → Privacy
- Security
- No need to rely on browser


Choose your passcode
This is the number you'll need to enter every time you use this app. It should be 5 to 8 digits long and not easily guessable.
Passcode
Confirm
Next

Apply proposed modification

- As stated in Design section
- To prove these can positively change

Q & A