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
Introduction

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
Introduction >> Background

Blockchain

Permissionless blockchain

- Proof-of-work
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 implementation
Related work
E-voting in Hong Kong

NO end-to-end verifiable system

Popvote

- Civil referendums
- Reports on security loopholes
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
E-voting using blockchain

1 vote = 1 coin

- Intermediate result
- Provable intention

Ballot as data

- Secure storage
Design
Overview

Helios - as reference
- Cryptography
- Limitation & Modification

Blockchain - as secure storage
- Type
- Protocol design
Cryptoraphy

Homomorphic El Gamal encryption

Create election

\[ p: \text{a prime number} \quad g: \text{a primitive root of } p \]

For each trustee:

private key: \( x_i, \ 0 < x_i < p - 1 \)  
public key: \( y_i = g^{x_i} \mod p \)

Election public key:

\[ y = y_1 y_2 y_3 \ldots \mod p \]

**Public:** \( \{p, g, y\} \)  
**Private:** \( \{x_1, x_2, x_3, \ldots\} \)
Prepare ballot

For each option in each question:

For each option in each question:

if voter choose this option, \( i = 1 \); else \( i = 0 \)

\[ m = g^i \mod p \]

random number: \( r \), \( 0 < r < p - 1 \)

\[ c_1 = g^r \mod p \quad \quad \quad \quad c_2 = y^r m \mod p \]

Encrypted option: \( \{c_1, c_2\} \)
**Cryptography**

**Compute result**

For each option in each question:

**Aggregation:**

$$c_1 = c_{1,1}c_{1,2}c_{1,3}... \mod p$$

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

$$c_2 = c_{2,1}c_{2,2}c_{2,3}... \mod p$$

**Decryption:**

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

$$g^m = c_2 (c_1 ^{x_1})c_1 ^{x_2}c_1 ^{x_3}... \mod p$$

**Result:** $m$ (discrete logarithm 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 logarithm

⇒ 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 $\propto$ Election scale
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
Design >> Blockchain

**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 >> Blockchain

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
Design >> Blockchain

Block generation

Node selection

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

<table>
<thead>
<tr>
<th>IP</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.1</td>
<td>3000</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>3001</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>5000</td>
</tr>
<tr>
<td>127.0.0.2</td>
<td>3000</td>
</tr>
<tr>
<td>127.0.0.3</td>
<td>3000</td>
</tr>
</tbody>
</table>

Ballot #1
Hash = 6789

6789 mod 5 = 4
4 + 1 = 5
Block generation

Consensus

- Byzantine Fault Tolerance algorithm
Block generation

Block broadcasting

- \( > \frac{1}{2} \) 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
Conclusion

Summary

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

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
- ...

Conclusion >> Planned work
User interface

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

Mobile application
- Personal device → Privacy
- Security
- No need to rely on browser
Apply proposed modification

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