D-voting
Final Presentation
What is D-Voting?

3 teams:
- Frontend
- Production-Readiness
- Security Audit
Front-end

Ahmed Elalamy, Ghita Tagemouati, Khadija Tagemouati
Introduction

Subject  Rank  Select  Text
Goals
Goals

Usability
Goals

Usability

Hint  Individual results
Goals

Usability

Robustness
Goals

Usability

Robustness

Authorization mechanism
Goals

Usability  Robustness  Accessibility
Goals

Usability  Robustness  Accessibility

Internationalization
Features & Implementations
Authorization Mechanism
Authorization Mechanism

How to make sure that only the form creator can manage it?
Authorization Mechanism

Roles

BEFORE
Authorization Mechanism

Roles → Policies
Authorization Mechanism

p, "SCIPER", "subject", "action"
Authorization Mechanism

AFTER casbin
Authorization Mechanism

```
p, 330361, roles, list
p, 330361, election, create
p, 330361, roles, remove
p, 330361, roles, add
p, 330361, proxies, post
p, 330361, proxies, put
p, 330361, proxies, delete

p, 175129, roles, list
p, 175129, election, create
p, 175129, roles, remove
p, 175129, roles, add
p, 175129, proxies, post
p, 175129, proxies, put
p, 175129, proxies, delete
```
Authorization Mechanism

```javascript
function isAuthorized(sciper: number | undefined, subject: string, action: string): boolean {
    return enf.enforceSync(sciper, subject, action);
}
```
Authorization Mechanism

More than sixty lines!
Authorization Mechanism

<table>
<thead>
<tr>
<th>id</th>
<th>ptype</th>
<th>v0</th>
<th>v1</th>
<th>v2</th>
</tr>
</thead>
<tbody>
<tr>
<td>character varying (256)</td>
<td>character varying (256)</td>
<td>character varying (256)</td>
<td>character varying (256)</td>
<td>character varying (256)</td>
</tr>
<tr>
<td>p</td>
<td>330383</td>
<td>roles</td>
<td>list</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>330383</td>
<td>roles</td>
<td>remove</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>330383</td>
<td>roles</td>
<td>add</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>330383</td>
<td>proxies</td>
<td>post</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>330383</td>
<td>proxies</td>
<td>put</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>330383</td>
<td>proxies</td>
<td>delete</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>330383</td>
<td>election</td>
<td>create</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>330382</td>
<td>roles</td>
<td>list</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>330382</td>
<td>roles</td>
<td>remove</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>330382</td>
<td>roles</td>
<td>add</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>330382</td>
<td>proxies</td>
<td>post</td>
<td></td>
</tr>
</tbody>
</table>
Authorization Mechanism

How to make sure that only the form creator can manage it?
Authorization Mechanism

```javascript
app.put('/api/elections/authorizations', (req, res) => {
  if (!isAuthorized(req.session.userid, SUBJECT_ELECTION, ACTION_CREATE)) {
    res.status(403).send('Unauthorized');
    return;
  }
  const { FormID } = req.body;
  enf.addPolicy(String(req.session.userid), FormID, ACTION_CREATE);
});
```
Translation & internationalization

Extension of Front-end functionalities
Future Improvements
Future Improvements

- Improve programmers' experience
Future Improvements

- Improve programmers' experience
- Improve error handling
Future Improvements

- Improve programmers' experience
- Improve error handling
- Use a new authorization model
Future Improvements

- Improve programmers' experience
- Improve error handling
- Use a new authorization model
- Make individual results only available to admins
Production Readiness

Amine Benaziz, Albert Troussard
Production readiness

Pre-Project Observations

- Lack of testing in realistic conditions
- Need for refactoring
- Lack of documentation
- Need for more metrics
Production readiness

Goals

Usability

Robustness

Performance
Production readiness

**Usability**

**Running script**

- Script to quickly & easily set up the system
- Minimized complexity & automated setup steps
- Easily maintain the system & make changes as needed
Production readiness

Usability

Code metrics

- SonarCloud: Code Analysis Platform
- Code Coverage
- Complexity
- Bugs
- Security Vulnerabilities
- Identify areas that need testing & improvement
Production readiness

Robustness

Codebase quality

- Review code & identify areas for improvement
- Refactor code & reduce code smells
- Improve codebase quality & ensure system security and reliability
Production readiness

Robustness

Authorization Mechanism

- Fail-safe default mechanism based on Sciper numbers
- Give rights to other users (admins/operators)
- Improve security & ensure only authorized users have access
Production readiness

**Robustness**

**Unit Testing**

- Verify system functioning correctly
- Focus on backend of the system
- Ensure system security & reliability
Production readiness

**Robustness**

**Integration Testing**

- Ensure system functioning correctly and components working together
- Test different scenarios (crashing node, revote, null ballots)
- Identify potential issues before they become serious problems
Production readiness

**Robustness**

**Scenario Test**

- Simulates real-world election from start to finish
- Benchmarking system performance and reliability
Production readiness

Robustness

- Time spent on each part of election aligned with expectations
- Main flaw of scenario test is lack of realism - txn only sent when previous accepted, takes time
- Needed test to mimic real-life, when ballots can be sent to blockchain at any time

<table>
<thead>
<tr>
<th>Create form</th>
<th>Setup DKG</th>
<th>Open form</th>
<th>Casting 100 ballots</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.08s</td>
<td>3.56s</td>
<td>1.16s</td>
<td>138.39s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Close form</th>
<th>Shuffle ballots</th>
<th>Decrypt ballots</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.56s</td>
<td>31.3s</td>
<td>7.9s</td>
</tr>
</tbody>
</table>

- Simulation of sending 100 ballots to 10 nodes
Production readiness

Performance

Load testing

- Similar to the scenario test but more realistic.
- Difference is in the sending of transactions.
Production readiness

**Performance**

**Observation on the load test**

- The number of ballots the client perceived as being accepted did not match the number of ballots that were actually added to the blockchain.
- Investigation has to be done on the way transactions were handled
Previous Transaction System

Client → Proxy

Transaction

Proxy → Recording the Blockchain

Recording the Blockchain → Blockchain

Add transaction to the pool

Ok, transaction added to the pool

Blockchain → Recording the Blockchain

Check if transaction was included during the recording

Recording the Blockchain → Client

transaction included / declined
Production readiness

Transaction mechanism

Issue

Client
Proxy
Pool
Blockchain

Transaction
Timeout

NO
New Transaction System

Client → Proxy
Transaction → Add transaction to the pool → Ok, transaction added to the pool
create token using transactions infos
UNKNOWN, Token

loop [until Status==INCLUDED]
Token → Translate token to transactions infos
Check if the transaction is in the blockchain
create token using transactions infos
Status, Token
Blockchain
Analysis

% of votes included with 1 vote/s with various duration

% of votes included

duration (minutes)

1  5  10
Production readiness

Performance

Conclusion of the load test

- Not all ballots sent are casted
- number of transactions in the pool ≤ number of nodes
Production readiness

Future work

• More investigation on the issue
Production readiness

Future work

- More investigation on the issue
- Proxy splitting to be resilient to DOS
Proxy splitting

Before
Production readiness

Proxy splitting

Clients

Cheap authentication proxy

proxys
Production readiness

Future work

- More investigation on the issue
- Proxy splitting to be resilient to DOS
- Batch inclusion verification requests
Security audit

Chen Chang Lew (Zac)
Achieving Security Properties

Authorization

Confidentiality

Privacy

Integrity

Verifiability

Availability
Plan for the semester

- Improve product security
- Define every security boundaries Flow
- Understand Dvoting system
- Define Threat Model
- Add Basic security (snyk, security txt)
- Investigation on DKG and Shuffler
- Investigation on smart contract
- Investigation on backend
- Investigation on frontend
- Investigation on backend
Summarize

- Create a **Threat Model** for D-voting
- Apply security scanning tools **SNYK** (scan **third-party** libraries)
- Add **security txt** (for researchers to report vulnerabilities)
- **Inspect** the whole codebase (found 9 security issues and 11 tech debt)
- Adding **vote verifiability** design and planning
Threat 5

Didn't check for the maximum length of the form.
Threat 6

Election not able to review if anyone submit a fake vote
Threat 7

ShuffleThreshold ≠ nbrSubmissionsThreshold

ShuffleThreshold > f

nbrSubmissionsThreshold > 2f

f = # of malicious node
Threat 8

Logout will not clear the sessions in browser
Threat 9

Only Admin and operator can vote

```javascript
// Secure /api/evoting to admins and operators
app.use('/api/evoting/*', (req, res, next) => {
    if (!isAuthorized(req.session.userid, SUBJECT_ELECTION, ACTION_CREATE)) {
        res.status(400).send('Unauthorized - only admins and operators allowed');
        return;
    }
    next();
});
```
## Created Issue (Security)

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>Tags</th>
<th>Opened by</th>
<th>Last Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>#253</td>
<td>A user who is not an admin or operator cannot vote.</td>
<td>security, web backend</td>
<td>Flamewind97</td>
<td></td>
</tr>
<tr>
<td>#252</td>
<td>Logout will not clear all the sessions in the browser.</td>
<td>security, web frontend</td>
<td>Flamewind97</td>
<td></td>
</tr>
<tr>
<td>#251</td>
<td>ShuffleThreshold shouldn't be used as the nbrSubmissions threshold</td>
<td>security, smart contract</td>
<td>Flamewind97</td>
<td></td>
</tr>
<tr>
<td>#250</td>
<td>Election will not be able to reveal the result if anyone submits a fake vote.</td>
<td>security, smart contract</td>
<td>Flamewind97</td>
<td></td>
</tr>
<tr>
<td>#249</td>
<td>Frontend create form didn't check for the maximum length of the form</td>
<td>security, smart contract</td>
<td>Flamewind97</td>
<td></td>
</tr>
<tr>
<td>#248</td>
<td>A user can vote multiple times (count as multiple votes) in an election</td>
<td>security, web backend</td>
<td>Flamewind97</td>
<td></td>
</tr>
<tr>
<td>#247</td>
<td>All the election stages can be ignored by malicious node</td>
<td>security, web backend</td>
<td>Flamewind97</td>
<td></td>
</tr>
<tr>
<td>#246</td>
<td>The public/private key of the election is changed by the Adversary</td>
<td>security, web frontend</td>
<td>Flamewind97</td>
<td></td>
</tr>
<tr>
<td>#217</td>
<td>Denial of Service, Dkg public key will always return false if an adversary compromise one device.</td>
<td>security, dkg</td>
<td>Flamewind97</td>
<td>Nov 23, 2022</td>
</tr>
</tbody>
</table>
Created Issue (Tech Debt)

<table>
<thead>
<tr>
<th>Issue Title</th>
<th>Priority</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Debt - Remove point &amp; public key related in web backend since we didn’t use it.</td>
<td></td>
<td>#245 opened last month by Flamewind97</td>
</tr>
<tr>
<td>Technical Debt - shouldn’t use fingerprint function for pseudorandomness because it is not efficient.</td>
<td></td>
<td>#244 opened last month by Flamewind97</td>
</tr>
<tr>
<td>Technical Debt - change loop and sleep to channel + ctx timeout to increase readability of code.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technical Debt - Need refactor in dkg &amp; shuffler</td>
<td></td>
<td>#220 by Flamewind97 was closed last month</td>
</tr>
<tr>
<td>Technical Debt - Duplicate function getForm()</td>
<td></td>
<td>#219 opened on Nov 23, 2022 by Flamewind97</td>
</tr>
<tr>
<td>Technical Debt - Encrypt function in DKG &amp;&amp; Marshall ballot function in smart contract is not used anymore</td>
<td></td>
<td>#218 opened on Nov 23, 2022 by Flamewind97</td>
</tr>
<tr>
<td>Technical Debt - check lenAddrs before sending getPeerKey</td>
<td></td>
<td>#216 by Flamewind97 was closed on Dec 15, 2022</td>
</tr>
<tr>
<td>Technical Debt - variable name &quot;buff, formIDBuf, formIDBuf, formID&quot; not consistent</td>
<td></td>
<td>#214 opened on Nov 16, 2022 by Flamewind97</td>
</tr>
<tr>
<td>Technical Debt - unclear/wrong comment</td>
<td></td>
<td>#213 opened on Nov 16, 2022 by Flamewind97</td>
</tr>
<tr>
<td>Technical Debt - change legacy structure “CreateForm”</td>
<td></td>
<td>#212 opened on Nov 16, 2022 by Flamewind97</td>
</tr>
<tr>
<td>Technical Debt - verify signature before execute request</td>
<td></td>
<td>#210 by Flamewind97 was closed on Dec 15, 2022</td>
</tr>
</tbody>
</table>
Verifiability
# Verifiability

<table>
<thead>
<tr>
<th>Election details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User xxx</strong></td>
</tr>
<tr>
<td><strong>User xxx</strong></td>
</tr>
</tbody>
</table>
Future Work

- Conduct a review of Dela and Kyber crypto package to identify any additional security risks because we assume them to be safe in the security report.
- Review and add more security measure tools for d-voting to ensure that it meets current security best practices.
- Conduct regular security assessments to identify and address any new or emerging security threats.
- Patch the security issues and technical debts.
Thanks for your time!